

Reconstrucción de objetos en 3D mediante un sistema estéreo binocular



Grado en Ingeniería Informática

Trabajo Fin de Grado

Alumno: Iker Jáuregui Elso

Director: Miguel Pagola Barrio

Pamplona, 13 de noviembre de 2018

Contenido

1. Introducción.....	4
1.1 Reconstrucción 3D.....	4
1.2 Motivación y objetivos	5
2. Conceptos básicos	6
2.1 Modelo de la cámara.....	6
2.2 Parámetros de la cámara	7
2.3 Conceptos relacionados con un sistema binocular.....	9
3. Reconstrucción 3D	10
3.1 Adquisición de imágenes	10
3.1.1 Construcción del sistema estéreo binocular	10
3.1.2 Configuración del objetivo y de la cámara	12
3.2 Calibración.....	14
3.3 Rectificación de imágenes y corrección de la distorsión óptica	16
3.4 Detección de puntos.....	17
3.5 Correspondencia de puntos	17
3.6 Triangulación	18
4. Modelo desarrollado	19
4.1 Sistema de visión	19
4.2 Adquisición de imágenes	21
4.3 Calibración.....	25
4.4 Rectificación y corrección de la distorsión óptica	27
4.5 Detección de puntos.....	29
4.5.1 Mangueta	29
4.5.2 Pelota	34
4.5.3 Cajas de cartón.....	36
4.5.4 Escuadra metálica	37
4.6 Correspondencia	38
4.7 Triangulación	41
5. Resultados.....	41
5.1 Calibración.....	41
5.2 Mangueta	43
5.3 Pelota	45
5.4 Cajas de cartón.....	46
5.5 Escuadra metálica	47
6. Conclusiones.....	49
6.1 Conclusiones técnicas	49
6.2 Conclusiones personales	50
6.3 Posibles mejoras	50
6.4 Líneas futuras.....	50
7. Anexos.....	52
Anexo I	52
Anexo II	56
8. Bibliografía.....	58

1. Introducción

Desde finales del siglo XX hasta la actualidad, la revolución tecnológica que hemos vivido nos ha llevado a desarrollar computadoras y dispositivos cada vez más avanzados. Hoy en día contamos con procesadores capaces de realizar miles de billones de operaciones por segundo, y los últimos telescopios empleados por la NASA, como el HIRISE (*High Resolution Imaging Science Experiment*) [1], pueden distinguir objetos de un metro de largo en la superficie marciana. Pero dejando de lado la tecnología más puntera que poseen las grandes universidades, institutos u organizaciones del mundo, cabe destacar que incluso las personas de a pie pueden contar con potentes aparatos electrónicos para uso privado. Por tanto, gracias a los avances tecnológicos de los últimos años y a la investigación y dedicación de multitud de profesionales, técnicas computacionalmente exigentes son hoy en día viables.

La reconstrucción de objetos en 3D no es algo nuevo, pero con la capacidad de cómputo de los últimos procesadores y tarjetas gráficas del mercado, cualquiera es capaz realizar una reconstrucción de calidad y en un tiempo asumible. Vivimos en una sociedad altamente tecnológica, en la que prácticamente cualquier actividad de nuestras vidas está acompañada por algún tipo de monitor o pantalla, y ya que la visión 3D nos transmite una mayor sensación de realidad, queremos que la mayoría de nuestros dispositivos nos muestren representaciones visuales tridimensionales. Pero, dejando de lado nuestro capricho por el realismo, la reconstrucción 3D también puede ser empleada con fines más técnicos, como por ejemplo en el control de calidad de productos, en el análisis de grandes estructuras o terrenos (fotogrametría), o en los sistemas de navegación de robots autónomos. Por tanto, como podemos ver, las aplicaciones de esta técnica son innumerables. Pero, ¿qué es exactamente la reconstrucción 3D?

1.1 Reconstrucción 3D

La reconstrucción 3D es el proceso por el cual se consigue reconstruir o representar objetos tridimensionales a partir de información no tridimensional. Esta información puede ser proporcionada por distintos dispositivos, como cámaras, láseres u otro tipo de sensores. Existen dos tipos de métodos por los que se puede conseguir este fin: métodos activos y métodos pasivos [2].

Los métodos activos reconstruyen el objeto incidiendo directamente en éste, ya sea por medio de un puntero o palpador mecánico que obtiene las coordenadas del objeto, como hace, por ejemplo, la máquina de medición por coordenadas (*Coordinate-measuring machine*, o CMM, en inglés) [3], o por la proyección de ciertos patrones de luz sobre la superficie del elemento en cuestión (escáner de luz estructurada).

Por el contrario, los métodos pasivos simplemente miden la luz que proviene de la superficie del objeto, para luego, a través de técnicas de visión artificial, obtener la información tridimensional. Es decir, los métodos pasivos reconstruyen el objeto a partir de imágenes adquiridas, generalmente, por cámaras digitales. Es posible realizar la reconstrucción con una sola cámara (sistema mono) o con varias cámaras (sistema estéreo).

Además, existen dos tipos de reconstrucción 3D que se pueden lograr con un sistema de visión estéreo: la reconstrucción dispersa (*sparse 3D reconstruction*) y la reconstrucción densa (*dense 3D reconstruction*) [4]. Las principales diferencias que existen entre estos dos tipos de reconstrucción 3D son, por un lado, la cantidad de puntos reconstruidos, y por otro, la precisión o calidad del resultado obtenido. La reconstrucción dispersa suele ser más rápida y sencilla de realizar, pero tiene serias dificultades a la hora de representar grabados, muescas u otros sutiles detalles. La reconstrucción densa, en cambio, si bien es más lenta, dado que utiliza algoritmos de correspondencia automáticos que pueden ser computacionalmente costosos, es capaz de recuperar los más pequeños detalles. No obstante, la reconstrucción densa está lejos de ser perfecta, ya que suele producir una gran cantidad de ruido y espacios sin reconstruir [5].

1.2 Motivación y objetivos

El principal objetivo de este proyecto es medir la precisión que se puede conseguir a la hora de reconstruir objetos en 3D con un sistema estéreo de cámaras. En concreto, se trata de un trabajo de investigación realizado en la empresa de visión artificial Tasiva Vision S.L, cuyo interés es comprobar si se podría sustituir la anteriormente citada máquina de medición por coordenadas por un sistema estéreo de visión artificial.

Las CMM son capaces de realizar mediciones de entre centésimas de milímetro y micras de precisión, pero presentan una gran cantidad de puntos desfavorables que vamos a comentar a continuación.

En primer lugar, las CMM son dispositivos complejos y caros, que necesitan estar situados en una instalación aclimatada y apartada de la línea de producción donde se controle tanto la temperatura como la humedad del ambiente. La precisión de estas máquinas depende en gran medida de estos factores, por lo que a la hora de validar un producto, hay que esperar a que éste adquiera la misma temperatura a la que se encuentra la sala. Por otro lado, estos dispositivos no son capaces de validar el producto automáticamente y han de ser manipulados por operarios especializados. Por todo ello, el control de calidad mediante máquinas de medición por coordenadas es un proceso lento y caro, que solamente permite validar una pequeña muestra de la producción diaria y, por tanto, tener un control estadístico (y no real) de toda la producción.

En cambio, un sistema estéreo multicámara de visión artificial puede colocarse en la misma línea de producción y validar automáticamente todos los productos fabricados in situ. Esta alternativa es más barata que el uso de una CMM, ya que las cámaras son, en comparación, componentes mucho más baratos que la propia máquina y, además, no requieren de una instalación especial o de un operario que las maneje. Si bien es cierto que es difícil alcanzar una precisión de centésimas de milímetro o de micras mediante una reconstrucción 3D con un sistema estéreo, también lo es que, en una gran cantidad de casos, no es necesario alcanzar tales precisiones. En muchos casos una precisión de décima de milímetro es más que suficiente, y por ello, y por el resto de razones mencionadas anteriormente, en estos últimos años la tendencia está siendo sustituir las CMM por sistemas de visión artificial.

Como un primer estudio, se decidió realizar una reconstrucción 3D dispersa de varios objetos con un sistema estéreo de dos cámaras, es decir, un sistema binocular. Por tanto, en este trabajo se explica, por una parte, la base teórica y fases de la reconstrucción de objetos en 3D mediante un sistema estéreo binocular, y por otra parte, la configuración del sistema implementado y las herramientas y técnicas empleadas, así como un análisis de los resultados obtenidos.

2. Conceptos básicos

A lo largo de este trabajo se habla de multitud de conceptos relacionados con las cámaras y su geometría, por lo que es necesario hacer, al menos, una pequeña introducción de estos conceptos para facilitar la comprensión de los contenidos presentados.

2.1 Modelo de la cámara

Comencemos, pues, presentando el modelo de cámara más sencillo: la cámara estenopeica (en inglés, *pinhole camera*). Se trata de una cámara sin lente, con una pequeña apertura (del orden de décimas de milímetro) por donde entra la luz [6]. Pese a que las cámaras de hoy en día cuentan con objetivos y mecanismos para variar dicha apertura, su funcionamiento puede aproximarse al de una cámara estenopeica. Por tanto, este modelo sirve perfectamente como ejemplo para representar el funcionamiento y las características de las cámaras en general.

En este modelo, el recién mencionado orificio se conoce como el centro de la lente, el centro de la cámara, el centro de proyección o centro óptico [7]. Por él penetra la luz que proviene de los puntos de la escena que se está capturando y se proyectan a cierta distancia, en el denominado plano de la imagen. Las imágenes proyectadas en este plano están invertidas, así que en el proceso de adquisición las cámaras digitales vuelven a invertir la imagen para que se corresponda con la escena que hemos capturado. Esta última imagen se proyecta en el llamado plano de la imagen virtual, en frente del centro de proyección [8]. Ambos planos están separados del centro de proyección por la misma distancia y son prácticamente equivalentes, por lo que no es raro ver representaciones de este modelo en las que se utilice, por conveniencia, el plano de la imagen virtual como si fuese el plano de la imagen real.

La distancia entre el centro de la lente y el plano de la imagen se conoce como distancia focal de la lente y sirve para establecer la relación entre cualquier punto de la escena, en coordenadas del mundo real (X, Y, Z), y el punto generado en el plano de la imagen (x, y).

En cuanto a los ejes de la cámara, tenemos, por un lado, los ejes X (eje horizontal) e Y (eje vertical), que forman un plano paralelo al plano de la imagen y, por otro lado, el eje Z , perpendicular a los dos anteriores y que pasa por el centro de la lente. Este último eje también se conoce como eje óptico.

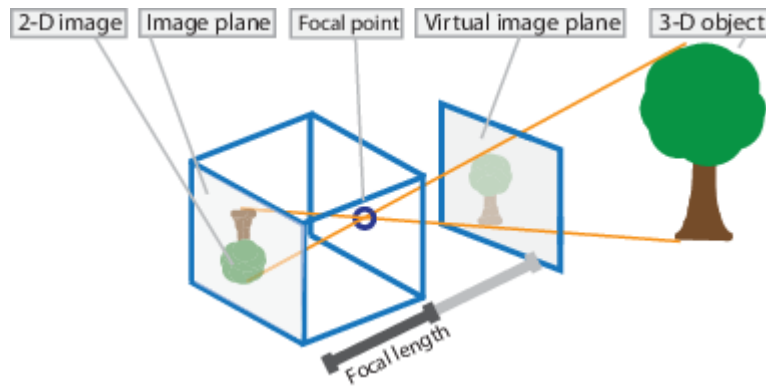


Figura 1: Modelo de la cámara estenopeica [8]

2.2 Parámetros de la cámara

Una vez aclarados los conceptos principales referentes al modelo y geometría de la cámara, veamos la información que se almacena y se utiliza de éstas. Existen dos tipos de parámetros que conciernen a la cámara, los parámetros intrínsecos y extrínsecos, ambos imprescindibles a la hora de realizar una reconstrucción 3D de calidad. Además, existe un tercer grupo de parámetros, conocidos como coeficientes de distorsión, que no son propios de la cámara sino más bien de la lente, pero que también juegan un papel determinante.

En primer lugar, los parámetros intrínsecos describen propiedades geométricas de la cámara, como la distancia focal y el centro óptico de la cámara [9]. Se utilizan para establecer la relación entre las coordenadas de la cámara y las coordenadas de la imagen. Se trata de una matriz cuadrada de 3x3, en la que los dos primeros elementos (f_x y f_y , respectivamente) de la diagonal principal representan la distancia focal en el eje horizontal y vertical de la imagen. En la mayoría de sensores, estos dos parámetros tienen el mismo valor, pero pueden encontrarse casos en los que sean diferentes. Los dos primeros elementos de la última columna (c_x y c_y), en cambio, describen las coordenadas en las que el eje óptico intersecciona con el plano de la imagen. Dicho punto se conoce como punto principal y, aunque parezca algo extraño, no suele estar en el centro de la imagen exactamente. Todos los parámetros intrínsecos vienen dados en píxeles, para que sus valores no cambien al escalar uniformemente la geometría de la cámara.

En segundo lugar, los parámetros extrínsecos se representan en una matriz de 3x4, dónde la submatriz de 3x3 tomada de la izquierda es una matriz de rotación y el vector 3x1 restante de la derecha es un vector de traslación. Pese a que sea algo contra-intuitivo, la matriz de parámetros extrínsecos describe como el mundo se transforma respecto a la cámara, por lo que se utiliza para mapear los puntos en coordenadas del mundo real en puntos en coordenadas de la cámara.

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

Figura 2: Matriz de parámetros intrínsecos (izquierda) y matriz de parámetros extrínsecos (derecha). La submatriz de rotación aparece resaltada en rojo y el vector de traslación en verde [9].

Por último, tenemos los coeficientes de distorsión. Las lentes de los objetivos de las cámaras pueden tener, principalmente, dos tipos de distorsión óptica: distorsión radial y distorsión tangencial [10].

La distorsión radial es la más común y surge por la propia forma de la lente. Esta distorsión, a su vez, puede ser de dos tipos distintos. Si la imagen está distorsionada en forma de barril (*barrel distortion*), entonces se trata de una distorsión radial positiva. Por el contrario, si la distorsión generada estira las esquinas de la imagen hacia afuera, en forma de cojín (*pincushion distortion*), se conoce como distorsión radial negativa. El número de coeficientes considerados para este tipo de distorsión dependen de la implementación del algoritmo empleado o de la precisión que se quiera alcanzar, pero siempre han de tenerse en cuenta al menos los dos coeficientes principales, conocidos como k_1 y k_2 . El resto de coeficientes (k_3, k_4, k_5, \dots) suelen considerarse despreciables en muchos casos, por lo que en algunos artículos ni siquiera aparecen reflejados.

La distorsión tangencial, en cambio, aparece cuando la lente no está perfectamente alineada con el sensor. Suele ser menos notoria que la anterior, pero también debe corregirse. En este caso, son dos los principales coeficientes considerados, denotados como p_1 y p_2 . Al igual que en la distorsión radial, los coeficientes de mayor orden (p_3, p_4, p_5, \dots) suelen despreciarse [11].

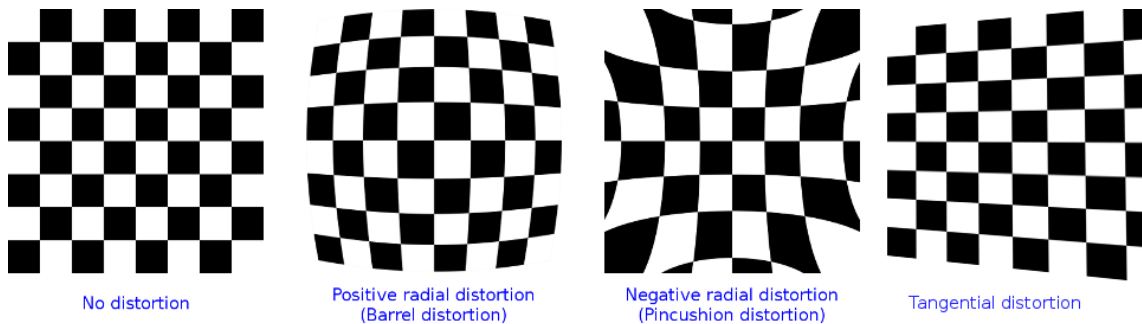


Figura 3: Ejemplos de las distintas distorsiones ópticas. De izquierda a derecha, imagen sin distorsión óptica, imagen con distorsión radial positiva, imagen con distorsión radial negativa e imagen con distorsión tangencial [8].

2.3 Conceptos relacionados con un sistema binocular

Tras conocer algunos detalles importantes de la cámara, centrémonos ahora en un par nociones básicas propias de un sistema estéreo binocular.

Por una parte, debemos introducir el concepto de solape. El solape es el área o campo de visión común entre las dos cámaras. Según la posición y orientación de los dispositivos, es posible que tengan una zona de la imagen más grande o más pequeña en común. Para poder realizar una reconstrucción tridimensional es imprescindible que ambas cámaras vean los mismos puntos que se van a reconstruir, ya que al adquirir una imagen se pierde la profundidad de la escena y es necesaria la redundancia de información para poder estimar la dimensión perdida.

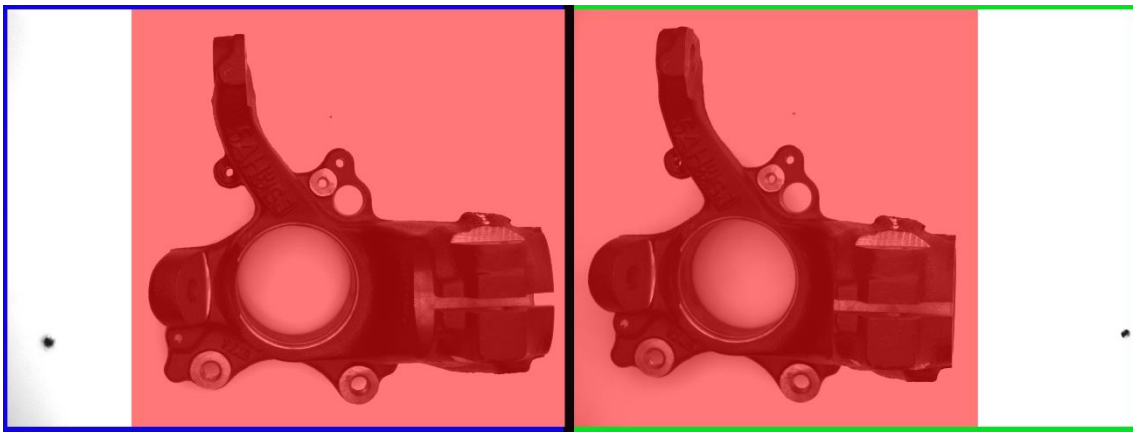


Figura 4: Par binocular de imágenes. La imagen de la cámara izquierda aparece resaltada en azul y la imagen de la cámara derecha en verde. El área resaltada en rojo es el solape entre ambas cámaras.

Por otra parte, tenemos la distancia entre los centros ópticos de las cámaras. Esta distancia se conoce como línea base (*baseline*, en inglés) [12] y juega un papel crucial en el proceso de triangulación y, por ende, en la reconstrucción 3D.

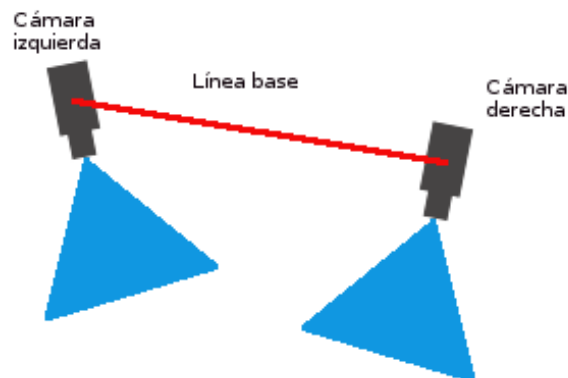


Figura 5: Sistema estéreo binocular. La línea roja representa la línea base.

3. Reconstrucción 3D

Ahora que hemos aclarado algunos conceptos, podemos comenzar a explicar las distintas fases de la reconstrucción 3D.

3.1 Adquisición de imágenes

Una de las partes más importantes y, a menudo, más obviadas de la visión artificial es la adquisición de imágenes. Puede parecer evidente, pero sin imágenes no puede haber visión artificial. Además, la calidad de las imágenes a analizar está altamente relacionada con la calidad o precisión del resultado que podemos obtener al utilizar técnicas de visión artificial. Por consiguiente, una buena configuración y disposición tanto de las cámaras como de los objetivos que éstas emplean, así como el uso de una iluminación adecuada, puede ser vital a la hora de conseguir los resultados que queremos. Por esta razón, en esta sección vamos a explicar brevemente algunos aspectos importantes de configuración y distribución de las cámaras.

3.1.1 Construcción del sistema estéreo binocular

Para empezar, debemos decidir cómo construir nuestro sistema estéreo binocular. Existen, principalmente, dos tipos de disposiciones a emplear, cada una con sus respectivas ventajas y desventajas.

En primer lugar, tenemos el denominado modelo estándar [13]. En esta organización, ambas cámaras tienen todos sus ejes paralelos entre sí y cuentan con la misma distancia focal. La única diferencia que existe entre las cámaras es que están separadas a cierta distancia en el eje horizontal (línea base). Esta estructura tiene como ventaja que, a la hora de calibrar el sistema estéreo y rectificar las imágenes, la propia disposición de las cámaras facilita el cálculo de las diferencias entre las perspectivas. Sin embargo, según cómo sea el objeto que estemos analizando, es posible que tengamos oclusiones o grandes diferencias de perspectiva, por lo que, en algunos casos, puede llegar a ser difícil o incluso imposible detectar todos los pares de puntos necesarios para realizar la reconstrucción 3D. Además, por lo general, con esta configuración los ángulos de intersección entre las líneas de los campos visuales de las cámaras suelen ser pequeños. Esto afecta negativamente a la precisión que se puede obtener al realizar la reconstrucción 3D.

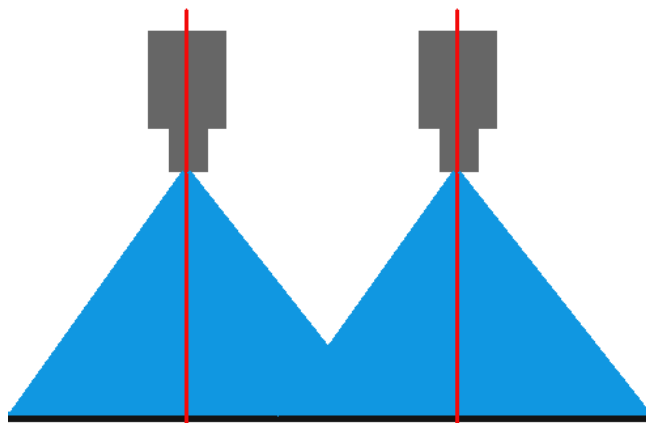


Figura 6: Modelo estándar de un sistema binocular. Las líneas rojas representan el eje óptico de las cámaras.

En segundo lugar, contamos con una ligera variante del modelo anterior. Esta vez, los ejes ópticos de las cámaras no son paralelos, sino que se encuentran algo inclinados de tal forma que existe una intersección entre ambos. En esta disposición se pierde la ventaja de la configuración anterior, por lo que es imprescindible rectificar las imágenes para poder hacer tanto la correspondencia como la triangulación de los puntos. No obstante, presenta algunos puntos favorables.

Por un lado, esta estructura genera un mayor solape entre las dos imágenes reduciendo las diferencias de perspectiva y favoreciendo la detección de puntos. Por otro lado, se puede conseguir un mayor ángulo de intersección entre las líneas de los campos visuales de las cámaras, lo que puede aumentar la precisión de nuestra reconstrucción tridimensional [14]. Aun así, cabe destacar que si el ángulo de intersección es grande, la precisión también puede verse afectada negativamente. Si se desea tener una buena precisión en este aspecto a lo largo del área de solape de ambas cámaras y, por tanto, en la zona que se puede reconstruir, se debe tener en cuenta que el ángulo de intersección entre las líneas de los campos visuales que inciden en el área de solape debe aproximarse a los 90° . Por lo que no es tan sencillo como disponer las cámaras de tal forma que los ejes ópticos generen un ángulo de 90° . Según cómo sea el área de solape, el objeto a reconstruir y los ángulos de visión de las cámaras (los ángulos del campo de visión, el ángulo en el eje X y el ángulo en el eje Y), el ángulo óptimo puede variar en gran medida.

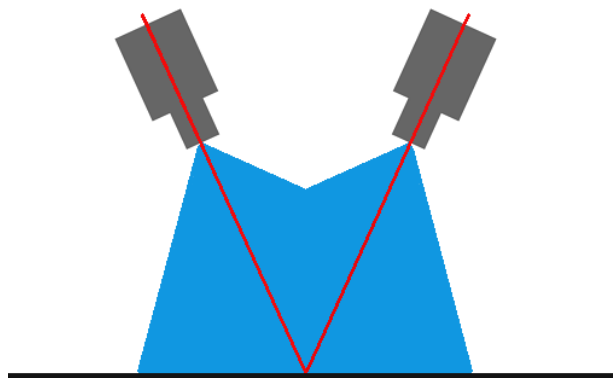


Figura 7: Sistema binocular con ejes ópticos cruzados. Las líneas rojas representan los ejes ópticos de las cámaras.

Por tanto, a la hora de construir un sistema binocular, es recomendable escoger la disposición que mejor se adapte a nuestro caso. Cabe destacar que, en la práctica, a menos que se empleen herramientas de medición o calibración muy precisas, es casi imposible lograr que tanto los ejes como la posición de las cámaras queden alineados perfectamente. Por suerte, mediante el proceso de calibración podemos averiguar la traslación y rotación que hay de una cámara a otra, para luego emplear esta información en la rectificación de las imágenes.

3.1.2 Configuración del objetivo y de la cámara

Tras escoger una disposición adecuada para nuestro sistema estéreo, debemos ajustar los objetivos de nuestras cámaras y configurar ciertos parámetros de adquisición para obtener imágenes de la mayor calidad posible. Se trata de un trabajo que depende enormemente del entorno y de las condiciones en las que vamos a realizar la reconstrucción 3D, por lo que en este apartado explicaremos algunos conceptos básicos de adquisición que son válidos para cualquier caso.

En primer lugar, necesitamos obtener una imagen nítida. Para ello, debemos ajustar el enfoque del objetivo de la cámara hasta que logremos la nitidez deseada. Es posible que no logremos tener bien enfocada la imagen en todos los puntos que queramos, ya sea porque estemos enfocando un objeto relativamente voluptuoso o porque parte del objeto se encuentre cerca de los bordes de la imagen. En estos casos necesitamos una mayor profundidad de campo, es decir, que el foco se mantenga a lo largo de una mayor distancia. Para lograrlo, debemos ajustar otra de las partes más importantes del objetivo: la apertura del diafragma.

El diafragma es el componente que controla la cantidad de luz que entra en la cámara. Manipulando su apertura podemos controlar la luminosidad de las imágenes, pero también la profundidad de campo del objetivo. Cuanto más abierto esté, más luz entrará y menor profundidad de campo tendremos. Por el contrario, cuanto más cerrado esté, más oscura serán nuestras imágenes pero mayor será la profundidad de campo.

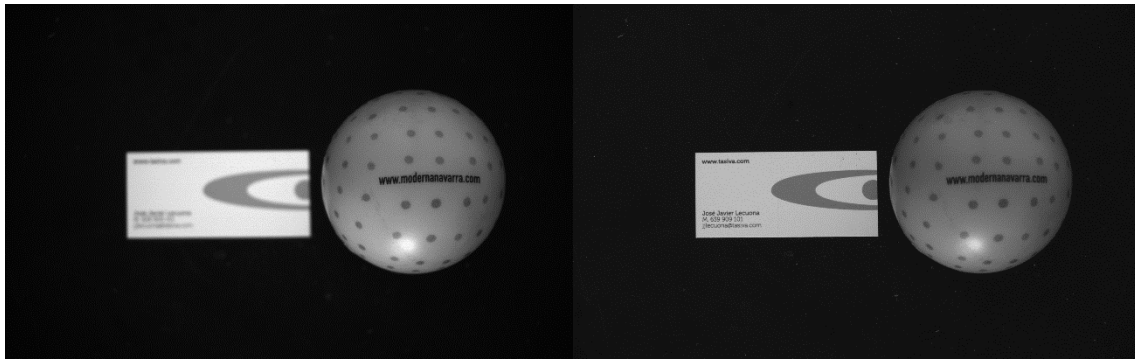


Figura 8: Ejemplo de profundidad de campo. La imagen de la izquierda no tiene suficiente profundidad de campo como para que tanto la pelota como la tarjeta de contacto se vean nítidas. Tras aumentar la profundidad de campo (derecha), ambos elementos aparecen nítidos.

Una vez ajustados el foco y el diafragma de los objetivos, debemos establecer el tiempo de exposición de las cámaras. Como su propio nombre indica, el tiempo de exposición es el tiempo en el que el sensor de la cámara está expuesto a la luz, es decir, el tiempo que la cámara está capturando la imagen. La adquisición de una imagen no es instantánea, la luz ha de estimular el sensor de la cámara a lo largo de un tiempo para que se forme la imagen. Una vez más, existe una especie de compuerta que controla durante cuánto tiempo entra la luz en el sensor, conocida como el obturador. Cuanto más tiempo permanezca el obturador abierto, más luz entrará en el sensor y, por tanto, más clara saldrá la imagen. Por el contrario, cuanto menor sea el tiempo de exposición, menor será la luz que entre en el sensor y la imagen será más oscura.

Se podría decir que estos tres conceptos son clave a la hora de adquirir imágenes. Como hemos dicho antes, la configuración y ajuste de la adquisición de imágenes es sumamente específica de cada caso. Si nos encontramos en una situación en la cual el objeto en cuestión se encuentra estático, quizás podamos abusar un poco del tiempo de exposición para obtener imágenes más luminosas. Sin embargo, si la captura se realiza en movimiento, debemos tener tiempos de exposición bajos para que las imágenes no salgan “movidas”, por lo que quizás tengamos que suplir la falta de luz con focos que iluminen la escena. Hay una inmensa cantidad de factores que pueden afectar al proceso de adquisición, además de múltiples parámetros de adquisición adicionales que pueden variar con los diferentes tipos de sensores de las cámaras. No obstante, los aspectos mencionados en este apartado son comunes a todos los casos y un buen ajuste de los mismos es imprescindible para cualquier situación.

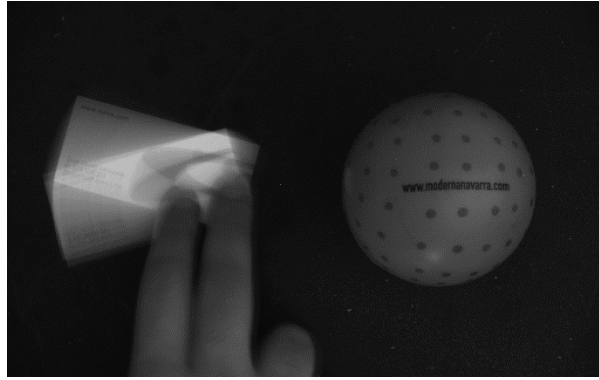


Figura 9: Ejemplo de imagen “movida”. Al tener un tiempo de exposición alto (1 segundo, en este caso), el elemento móvil de la imagen (tarjeta de contacto) aparece emborronado y en varias posiciones al mismo tiempo, mientras que el elemento estático (pelota) se ve correctamente.

3.2 Calibración

Tras construir nuestro sistema binocular y conseguir adquirir imágenes de calidad, llega el momento de comenzar con una parte esencial de la reconstrucción 3D: la calibración de las cámaras.

El proceso de calibración tiene dos objetivos principales. Por una parte, trata de encontrar los parámetros intrínsecos y extrínsecos de las cámaras y, por otra parte, trata de conocer los coeficientes de distorsión de la lente.

Para poder calcular toda esta información, es necesario capturar y analizar un objeto cuyas dimensiones y características sean conocidas. Por lo general, no se calibra con cualquier objeto, sino que se utiliza un patrón de calibración para ello. El patrón de calibración es, básicamente, una tabla o cualquier elemento plano y rígido que tenga un patrón en blanco y negro impreso sobre su superficie. Los hay de dos tipos principalmente: patrones de tablero de ajedrez y patrones de malla de círculos.

Los patrones de tablero ajedrez cuentan con una disposición alterna de cuadrados blancos y negros del mismo tamaño. Los patrones de malla de círculos, en cambio, presentan una disposición de círculos negros del mismo diámetro sobre un fondo blanco (o al revés, círculos blancos sobre un fondo negro). Los algoritmos que utilizan patrones de tablero de ajedrez utilizan la distancia entre las esquinas de los cuadrados para la calibración, mientras que los que utilizan los patrones de malla de círculos emplean la distancia entre los centros de éstos.

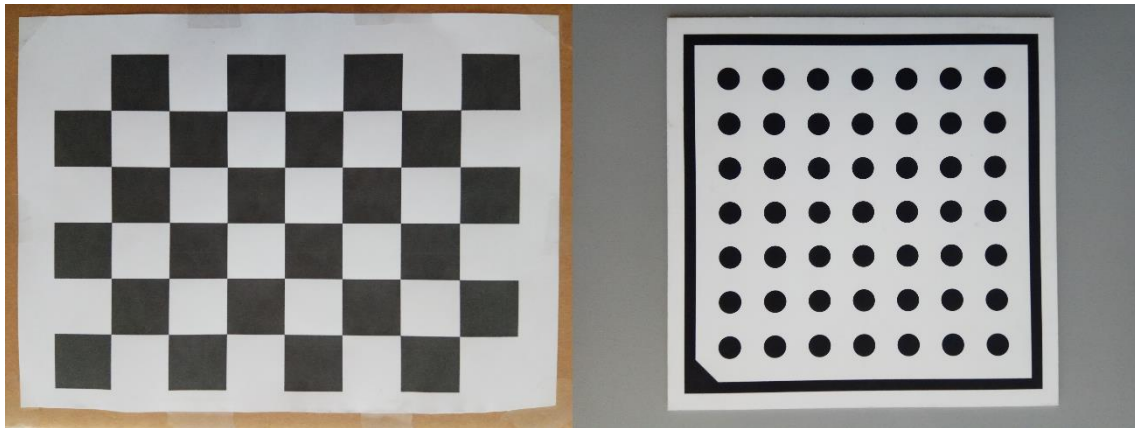


Figura 10: Ejemplos de patrones de calibración. A la izquierda, un patrón de tablero de ajedrez y, a la derecha, un patrón de malla de círculos.

En cuanto al tipo de imágenes que han de tomarse del patrón de calibración, es recomendable capturar imágenes de éste colocándolo a lo largo de todo el campo visual de la cámara. Además, se aconseja que se varíe la orientación del patrón, girándolo en distintas posiciones e incluso inclinándolo un poco por alguno de sus lados. También se sugiere que el tamaño del patrón de calibración sea lo más grande posible, para que en el proceso de calibración se pueda apreciar mejor la distorsión óptica generada por las lentes.

Respecto a la calibración de una cámara, existen varios métodos para ello, pero, al final, todos se reducen a resolver el sistema de ecuaciones generado al adquirir imágenes de un patrón de calibración [15]. Primero, tanto los puntos en coordenadas del mundo real como los puntos de la imagen se pasan a coordenadas homogéneas. Gracias a esta transformación conseguimos que las operaciones matriciales que se van a desempeñar cuadren perfectamente (además, no modifican en absoluto los valores reales).

No obstante, el mayor truco está en que, el patrón de calibración, al ser un objeto plano, los puntos detectados tienen la misma componente Z . O lo que es lo mismo, podemos establecer que el sistema de coordenadas del mundo real comience con el eje Z en el mismo plano en el que yace el patrón. Por tanto, el valor de Z de todos los puntos del patrón será igual a cero, lo que elimina la tercera columna de la matriz de rotación. De esta forma, para resolver el sistema de ecuaciones tan solo se necesitan tres imágenes, con al menos cuatro puntos en cada una de ellas [16]. Sin embargo, para minimizar el error generado, se suele tomar una mayor cantidad de imágenes.

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Figura 11: Ejemplo del sistema de ecuaciones a resolver. En rojo se señala cómo se simplifica el sistema al establecer a cero el componente Z de los puntos [16].

Respecto a la calibración binocular, se sigue un proceso semejante, aunque han de realizarse algunos pasos adicionales, ya que, además de calcular los parámetros intrínsecos y extrínsecos de ambas cámaras y calcular los coeficientes de distorsión de las lentes de sus respectivos objetivos, se debe estimar la rotación y traslación que existe entre una cámara y otra.

3.3 Rectificación de imágenes y corrección de la distorsión óptica

Una vez calibrado nuestro sistema binocular, podemos emplear la información conseguida para rectificar las imágenes adquiridas y corregir la distorsión óptica de las lentes. El proceso de rectificación consiste en conseguir que cada punto de una imagen aparezca alineado horizontalmente con el punto correspondiente de la otra imagen [17]. Al cumplir esta condición, limitamos el problema de la correspondencia de un punto a la búsqueda en una sola línea horizontal, además de simplificar el cálculo de la coordenada Z en el proceso de triangulación. Por tanto, se trata de un paso realmente importante de la reconstrucción 3D. En un caso ideal, si nuestro sistema binocular estuviese construido siguiendo el modelo estándar y las lentes no presentaran ningún tipo de distorsión, obtendríamos dos imágenes capturadas en el mismo plano y con la misma orientación. No obstante, como hemos visto antes, conseguir estas condiciones es realmente difícil y puede presentar algunos problemas. Sin embargo, mediante la rectificación de imágenes y la corrección de la distorsión óptica podemos lograr el mismo efecto [18], por lo que son, sin duda, pasos indispensables a la hora de realizar una reconstrucción 3D.

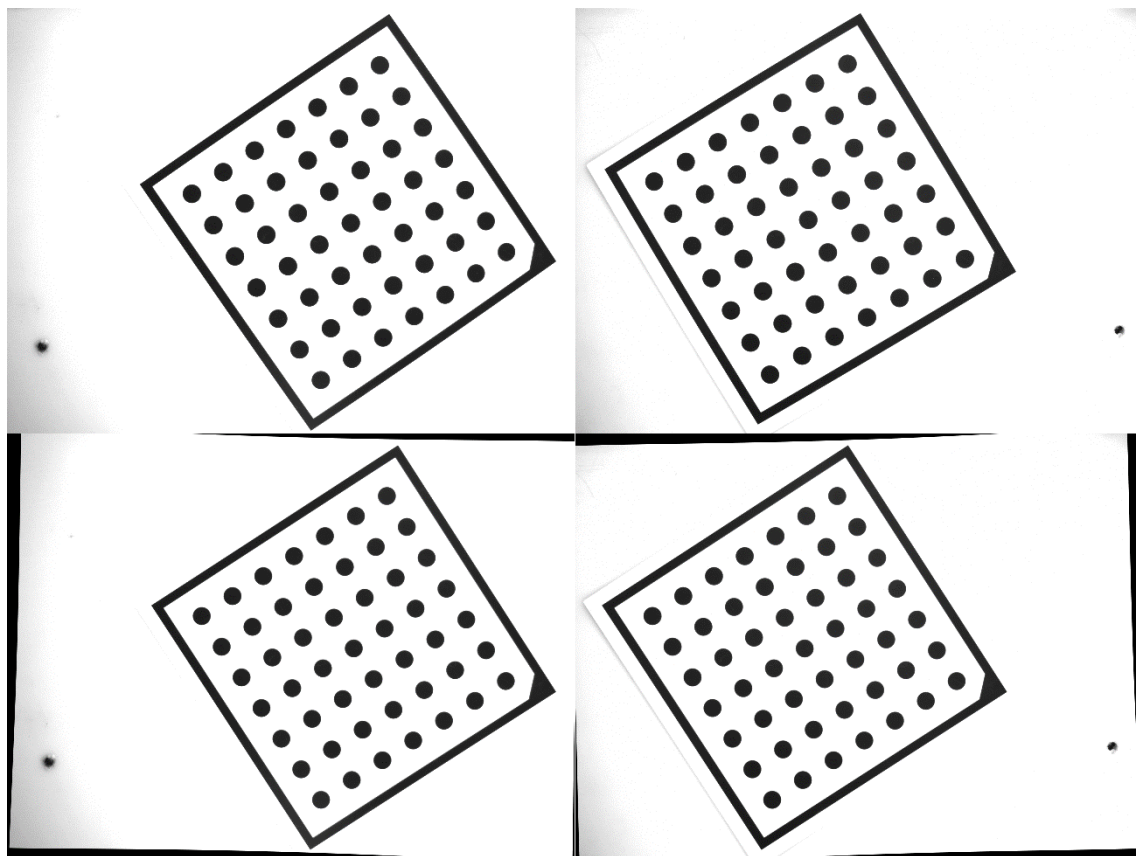


Figura 12: Arriba, par estéreo de imágenes de un patrón de calibración. Abajo, esas mismas imágenes tras ser rectificadas y corregir la distorsión óptica.

3.4 Detección de puntos

Tras conseguir que el par estéreo de imágenes esté rectificado y libre de la distorsión óptica generada por las lentes, llega el momento de seleccionar los puntos que se desean reconstruir en 3D. Las técnicas utilizadas para ello pueden ser muy diferentes según cual sea nuestro propósito. Existe un gran abanico de posibilidades, desde opciones poco elegantes, como la selección de los píxeles de forma manual, hasta el empleo de técnicas más sofisticadas, como algoritmos automáticos de detección de texturas. Sin embargo, sea cual sea la técnica empleada, siempre ha de cumplirse una condición: tanto en una imagen como en la otra debe haber la misma cantidad de puntos seleccionados. Recordemos que, para realizar la reconstrucción 3D de un punto, se precisa de redundancia de información. Es decir, al menos se necesita tener dos proyecciones de dicho punto desde distintas perspectivas. Por tanto, un punto de una imagen que no tiene su correspondiente pareja en la otra imagen no puede ser reconstruido.

3.5 Correspondencia de puntos

Después de escoger o detectar los puntos a reconstruir, debemos resolver el problema de la correspondencia. La correspondencia de puntos es, simplemente, establecer qué puntos de una imagen corresponden con los de la otra imagen [19]. Por tanto, la resolución de este problema reside en cómo se emparejan los puntos del par estéreo de imágenes. Una vez más, se nos presenta un gran abanico de posibilidades. Si la cantidad de puntos

detectados es relativamente baja, podemos resolver la correspondencia de forma manual o con simples algoritmos de ordenación. Por el contrario, si en el paso previo hemos empleado, por ejemplo, un algoritmo de detección de texturas, tendremos una gran cantidad de puntos que manejar. En este caso, seguramente necesitemos hacer uso de algún algoritmo de correspondencia que vaya minimizando alguna medida de error, ya sea de forma local (por bloques, por ejemplo) o de forma global. Existen una gran cantidad de medidas de error que se pueden utilizar, como, por ejemplo, el error cuadrático o el error absoluto, así como varios enfoques distintos con los que atacar el problema, por lo que no hay una forma estándar de resolver la correspondencia [20]. Una vez más, la mejor forma de resolver la correspondencia depende de cada caso, por lo que habrá técnicas y medidas que se adapten mejor a nuestros requisitos y otras que se adapten peor.

3.6 Triangulación

Una vez resuelto el problema de la correspondencia de puntos, podemos realizar la reconstrucción 3D. Como hemos visto anteriormente, al adquirir una imagen estamos generando una proyección bidimensional de una escena tridimensional. Por tanto, en el proceso de adquisición estamos perdiendo una dimensión de la escena capturada, o lo que es lo mismo, estamos perdiendo la percepción de profundidad de la escena. Si bien hemos comentado que para recuperar dicha profundidad necesitamos, al menos, dos imágenes de esa escena, no hemos explicado exactamente por qué. La razón es que la profundidad está estrechamente relacionada con otro concepto: la disparidad. De hecho, la profundidad es proporcionalmente inversa a la disparidad. Pero, ¿qué es la disparidad?

Pese a que el concepto de disparidad pueda resultarnos completamente nuevo, entenderlo no lo es tanto. La disparidad es, sencillamente, la diferencia de posición entre las dos proyecciones generadas en ambas imágenes de un mismo punto tridimensional de la escena [21]. Si contamos con que nuestras imágenes han sido perfectamente rectificadas, esta disparidad será, simplemente, la diferencia en el eje horizontal de las imágenes. Por tanto, valores grandes de disparidad indican que el punto de la escena se encuentra a una baja profundidad (o gran proximidad) y valores pequeños, en cambio, justo lo contrario.

Para una mejor comprensión de la relación entre la profundidad y la disparidad, podemos hacer un sencillo experimento. Si miramos a un punto en el infinito y colocamos nuestro dedo índice levantado en frente de nuestros ojos, veremos nuestro dedo dos veces. Eso es porque, al enfocar en el infinito y al tener el dedo tan cerca de los ojos, la diferencia de perspectiva entre éstos es muy grande en ese punto. O dicho de otra forma, la disparidad será muy grande. Sin embargo, a medida que alejemos el dedo de nuestra cara, la distancia entre los “dos” dedos se irá reduciendo, ya que disminuirá la disparidad.

Por tanto, acabamos de entender el concepto clave de la reconstrucción 3D mediante sistemas estéreo. No obstante, solamente con los valores de disparidad no podemos calcular la profundidad, necesitamos más datos. Uno de ellos es la distancia focal de las lentes y, el otro, la distancia de la línea base. Por suerte, mediante el proceso de calibración hemos obtenido esa información, por lo que, ahora sí, somos capaces de calcular la dimensión perdida.

Para ello, se emplea un método conocido como triangulación que calcula la profundidad (z) relacionando la disparidad (d), la distancia focal (f) y la línea base (b) de la siguiente manera [22]:

$$z = (f * b) / d$$

4. Modelo desarrollado

Tras presentar los conceptos básicos relacionados con un sistema estéreo binocular, además de los distintos pasos a seguir para realizar una reconstrucción tridimensional, en esta sección veremos los sistemas binoculares construidos, además de la tecnología, algoritmos y objetos empleados en las pruebas que se han llevado a cabo.

4.1 Sistema de visión

Para empezar, presentaremos los componentes empleados en la construcción de los sistemas estéreo, así como la configuración de los mismos.

Los sistemas desarrollados cuentan con dos cámaras “Basler ace acA2500-14gm”, con una resolución de 5 mega-píxeles (imágenes de 2592x1944 píxeles) e interfaz de comunicación “GigE Vision” (estándar muy extendido en la industria, para la transmisión de imágenes y vídeo por Ethernet) [23]. Llevan un sensor monocromo incorporado, por lo que las imágenes que adquieren son en escala de grises, es decir, en “blanco y negro”. Ambas cámaras están alimentadas por una fuente “Mean Well DR-75-12” de 12 voltios. En cuanto a los objetivos empleados, se tratan de objetivos “Basler Lens C125-0818-5M F1.8 f8mm”, con lentes de 8mm de distancia focal.



Figura 13: Componentes del sistema binocular implementado. En la esquina superior izquierda, las dos cámaras “Basler ace acA2500-14gm”. En la esquina inferior izquierda, los dos objetivos “Basler Lens C125-0818-5M F1.8 f8mm”. A la derecha, la fuente de alimentación “Mean Well DR-75-12”.

Respecto a la disposición de las cámaras, se han utilizado dos configuraciones distintas. Para realizar la reconstrucción de una pieza de automoción, llamada mangueta, se ha empleado la disposición estándar. Es decir, las cámaras se han dispuesto con la misma orientación, con la intención de que los ejes X, Y y Z sean paralelos entre una cámara y la otra. La distancia establecida entre el punto de sujeción de las cámaras y la mesa de trabajo es de 588’3mm.

En cuanto al resto de objetos a reconstruir (pelota, cajas de cartón y escuadra metálica), se ha empleado la estructura de ejes ópticos cruzados. Es decir, partiendo de la configuración anterior, se han girado las cámaras en el eje Y, aproximadamente, 12 grados, con el propósito de aumentar tanto el solape como el ángulo de intersección de los campos visuales de las cámaras. Esta vez, ya que los objetos a reconstruir son más pequeños, se ha disminuido la altura a la que están posicionadas las cámaras a 492mm respecto a la mesa de trabajo. En ambas configuraciones se ha utilizado la misma separación entre cámaras, siendo esta de 84mm.

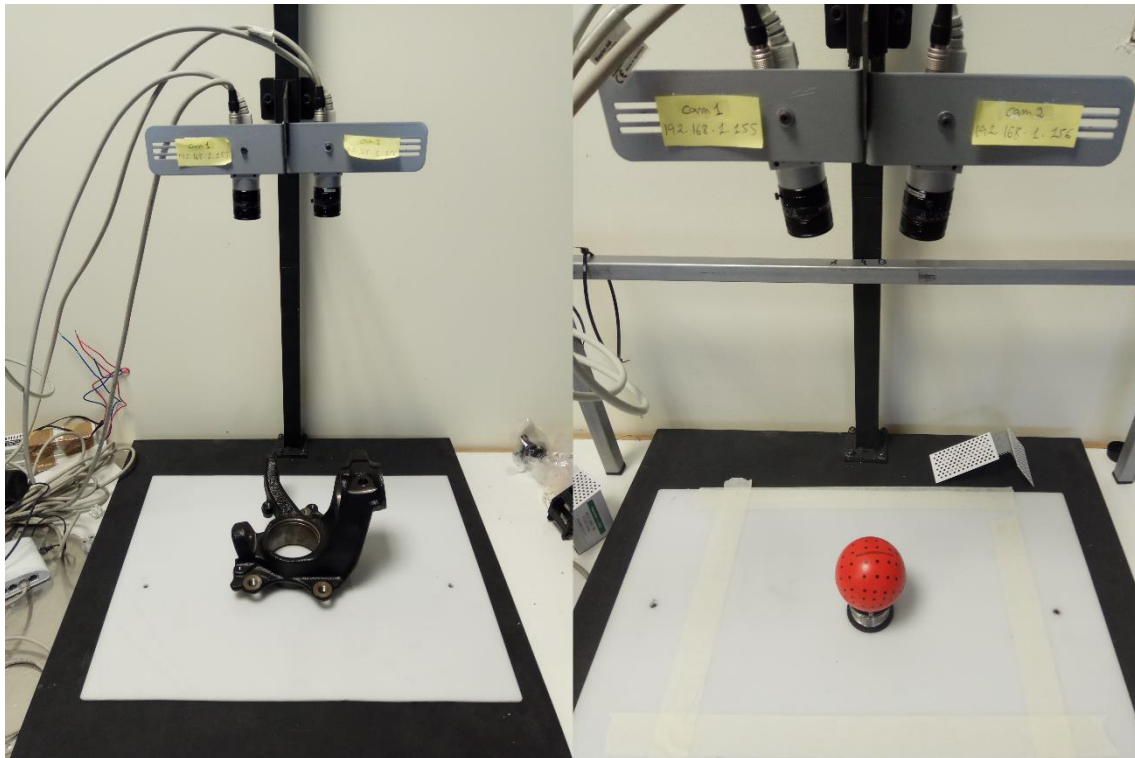


Figura 14: Sistemas binoculares implementados. A la izquierda, el sistema binocular estándar implementado para la reconstrucción de una mangueta (pieza de color negro). A la derecha, el sistema binocular de ejes ópticos cruzados construido para la reconstrucción de una pelota, una escuadra metálica y unas cajas de cartón.

4.2 Adquisición de imágenes

En cuanto a la adquisición de imágenes, se ha empleado el software de adquisición y configuración de cámaras llamado Pylon, que el propio fabricante de las cámaras (Basler) provee de forma completamente gratuita en su página web. En concreto, la versión utilizada es la 5.1.0 para sistemas operativos Linux de 64 bits. La herramienta se puede conseguir desde el siguiente enlace: <https://www.baslerweb.com/en/sales-support/downloads/software-downloads/>

Los tiempos de exposición empleados son de 230000 microsegundos para las imágenes capturadas con la cámara izquierda y 200000 microsegundos para las imágenes de la cámara derecha. La diferencia en los tiempos de adquisición se debe a que las pruebas realizadas no se han realizado en un entorno plenamente controlado, por lo que la iluminación ambiental afecta de diferente forma a cada cámara.

Respecto a las imágenes adquiridas, se ha capturado un par estéreo de cada objeto a reconstruir (Figura 15), o sea, por cada objeto, una imagen con la cámara izquierda y otra imagen con la cámara derecha. Además, para la calibración de las cámaras del primer sistema binocular implementado, el modelo estándar, se han tomado un total de 27 pares de imágenes del patrón de calibración (Figura 16). Para la calibración del sistema binocular de ejes ópticos cruzados, en cambio, se han adquirido un total de 47 pares de imágenes del patrón de calibración (Anexo I).

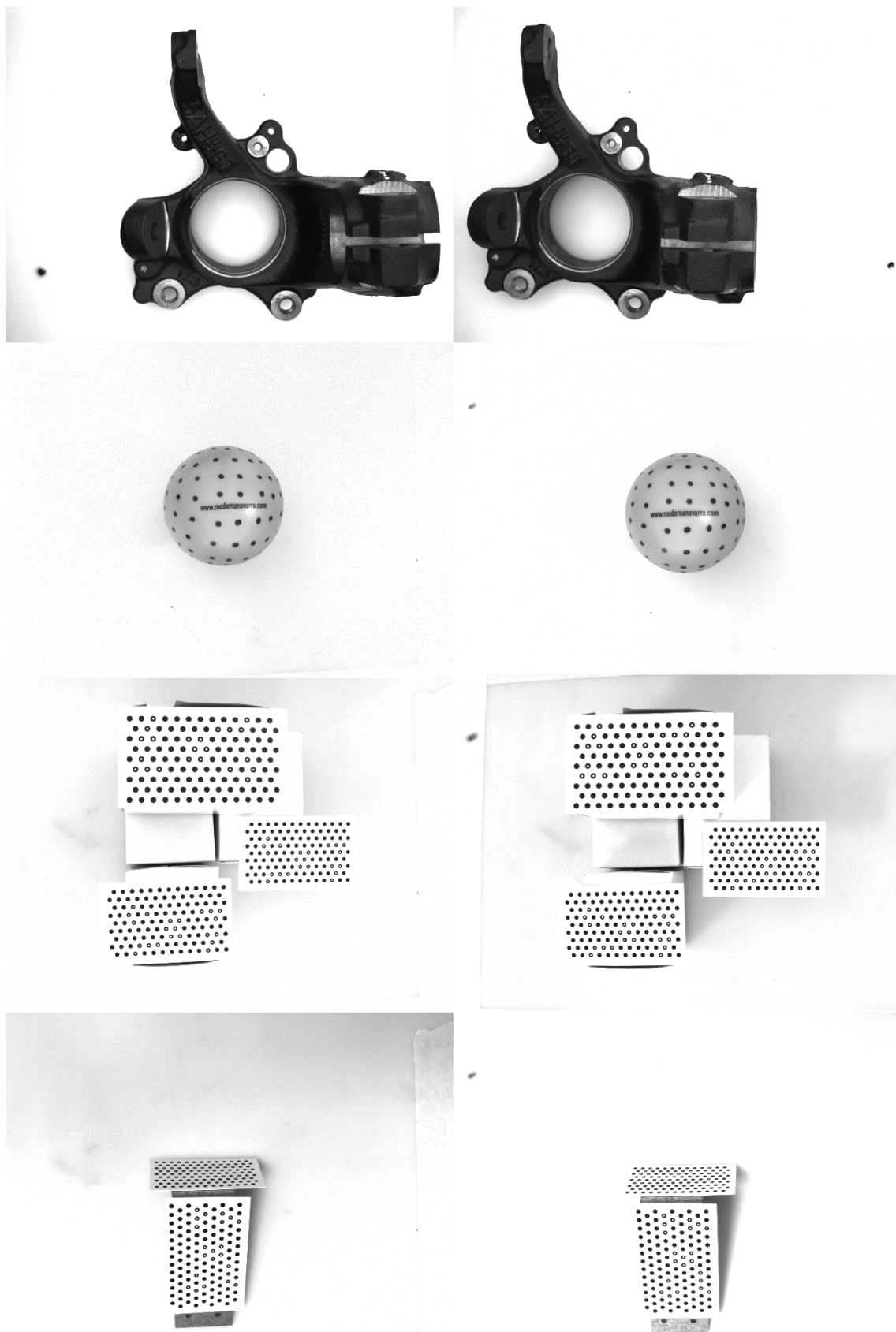
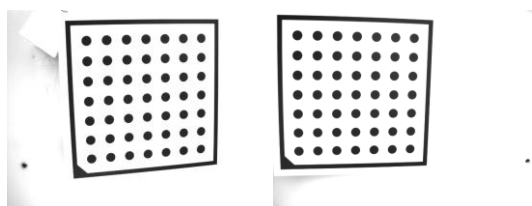
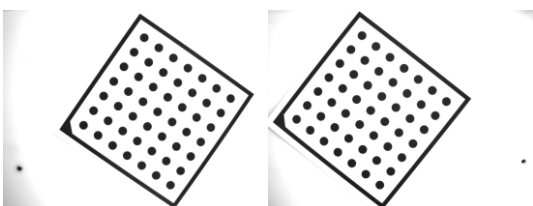
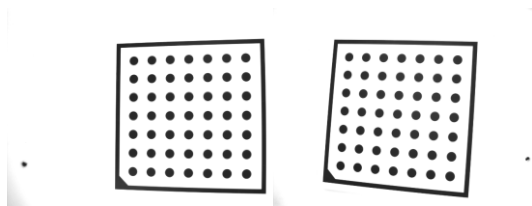
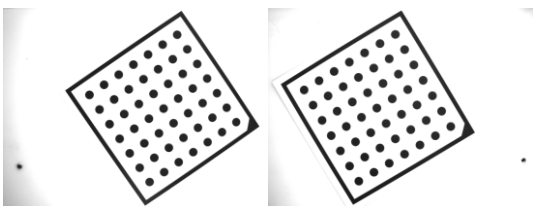
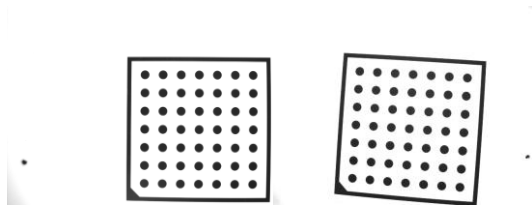
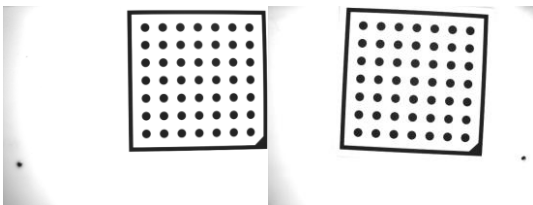
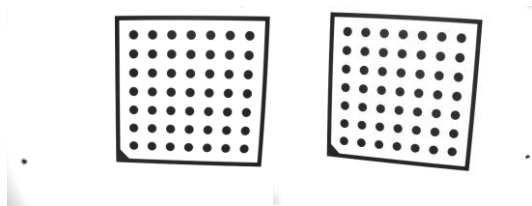
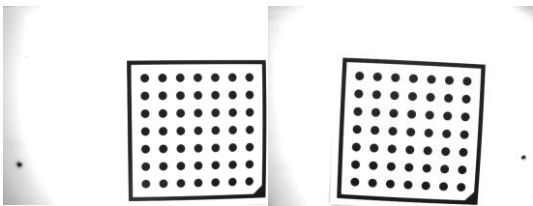
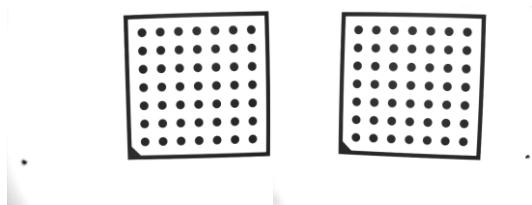
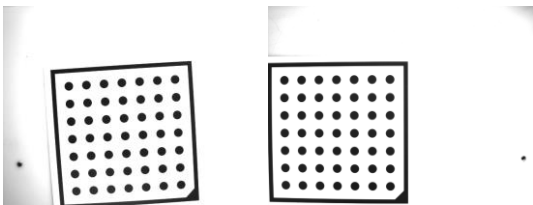
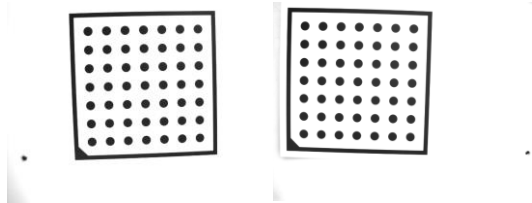
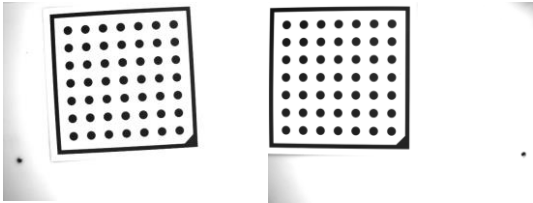
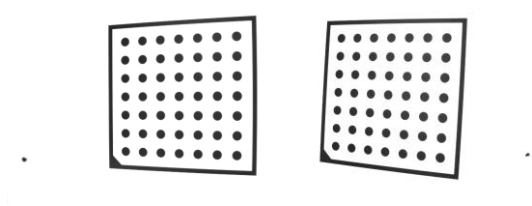
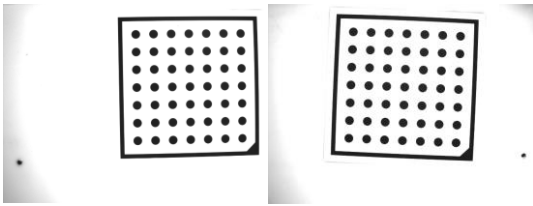


Figura 15: Pares estéreo de los objetos a reconstruir. En orden descendente: una mangueta, una pelota, unas cajas de cartón y una escuadra metálica.



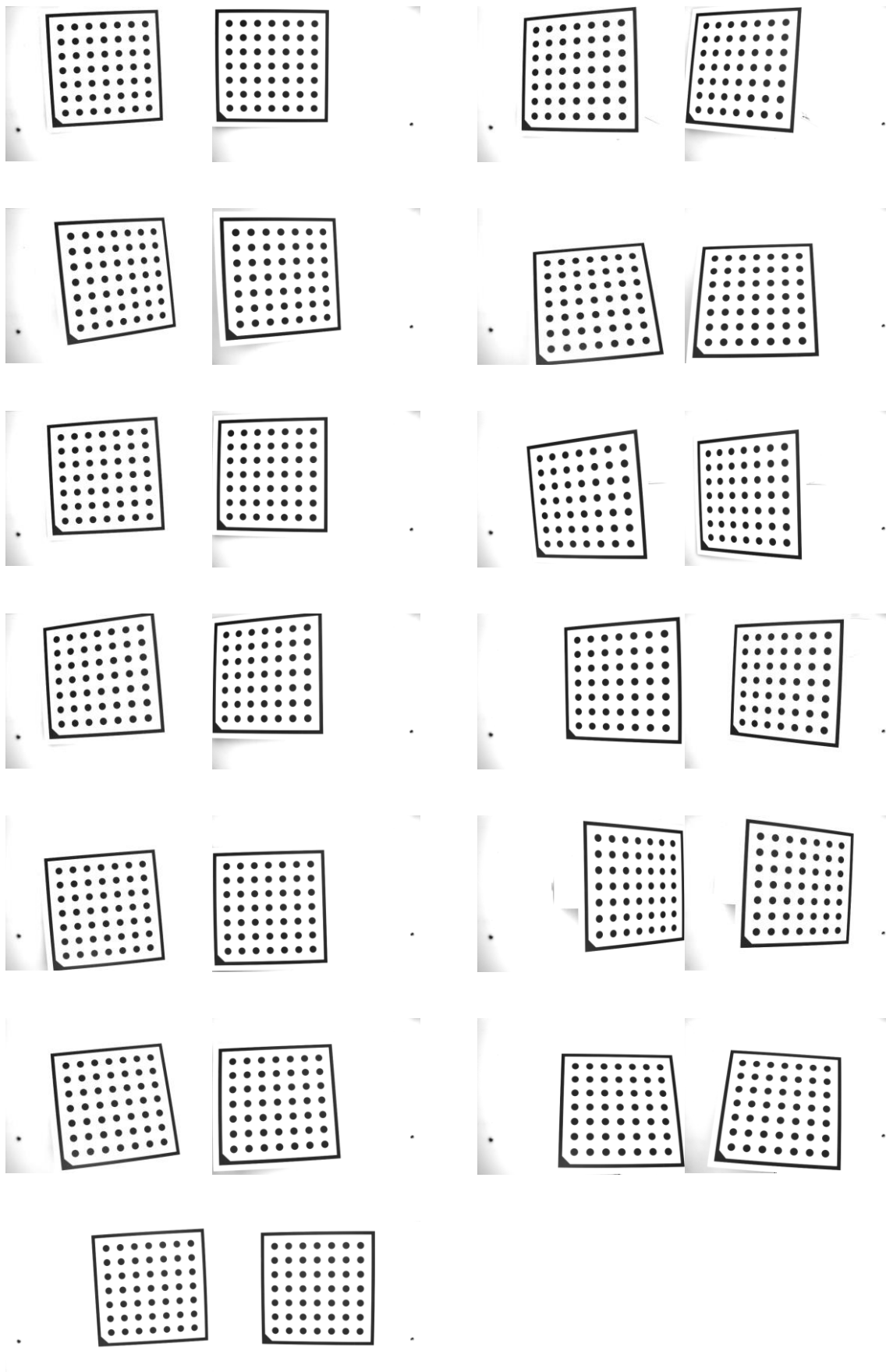


Figura 16: 27 pares estéreo del patrón de calibración para la configuración estándar.

4.3 Calibración

Tras presentar las dos distintas configuraciones del sistema binocular implementadas y las imágenes adquiridas, veamos el proceso de calibración que se ha llevado a cabo para calibrar las dos cámaras en ambas configuraciones. Cabe destacar que, tanto para el proceso de calibración como para el resto de procesos involucrados en la reconstrucción tridimensional (rectificación, corrección de la distorsión óptica, detección y correspondencia de puntos, y triangulación), se ha hecho uso de la librería de visión artificial OpenCV [24]. En concreto, se ha empleado la versión 3.2.0 implementada para el lenguaje de programación C++ y para sistemas operativos Linux de 64 bits. Además, para el proceso de calibración de los dos diferentes sistemas binoculares implementados, como se puede observar en la Figura 16 y en el Anexo I, se ha utilizado un patrón de calibración de malla de círculos de 7x7 (número de círculos en filas y en columnas), con una distancia entre los centros de 25mm.

Respecto a realizar una calibración con las herramientas que nos proporciona OpenCV, se trata de una tarea realmente sencilla. Simplemente debemos utilizar la función *stereoCalibrate* [25] que se encuentra disponible en el módulo *calib3d*. No obstante, esta función toma como parámetros de entrada algunos elementos de los que, a priori, no disponemos. Por una parte, toma como entrada los puntos del patrón de calibración en coordenadas del mundo real y, por otra parte, toma como entrada las proyecciones de estos puntos generadas por las dos cámaras. Sin embargo, por el momento, solamente contamos con los pares de imágenes del patrón de calibración. Por tanto, antes de emplear esta función, es necesario realizar un proceso previo de extracción de los puntos requeridos. Para ello, se ha implementado una versión del algoritmo publicado por Sourish Ghosh [26].

El funcionamiento de este algoritmo no es complejo, en absoluto. Se trata de un bucle iterativo, en el que, en cada iteración, se carga en memoria un par estéreo del patrón de calibración, para luego llamar a la función *findCirclesGrid* [27] con cada imagen del par. Esta función, devuelve como parámetro de salida una variable booleana. Si la variable de salida es verdadera, significa que la función ha sido capaz de encontrar todos los círculos del patrón de calibración. Si, por el contrario, es falsa, significa que la función no ha sido capaz de encontrar todos los círculos. En caso de éxito, obtendremos, también, un vector de puntos 2D con las coordenadas de los centros de los círculos.

De forma análoga al proceso de correspondencia de puntos, en este proceso previo al de calibración, es imprescindible que se hallen todos los círculos del patrón en ambas imágenes. En caso contrario, el par estéreo será rechazado y, de haberlo, se continuará con el siguiente par. Además, aunque se encuentren el mismo número de puntos, también es importante que aseguremos la correspondencia entre éstos. Para ello, podemos visualizar los puntos detectados con la función *drawChessboardCorners* [28] (que, pese a su nombre, también puede ser utilizada con un patrón de calibración de malla circular). Cuando hayamos tenido éxito en la búsqueda de los círculos, debemos almacenar los vectores de puntos 2D conseguidos, ya que éstos son las proyecciones de los puntos que la función *stereoCalibrate* demanda como parámetros de entrada.

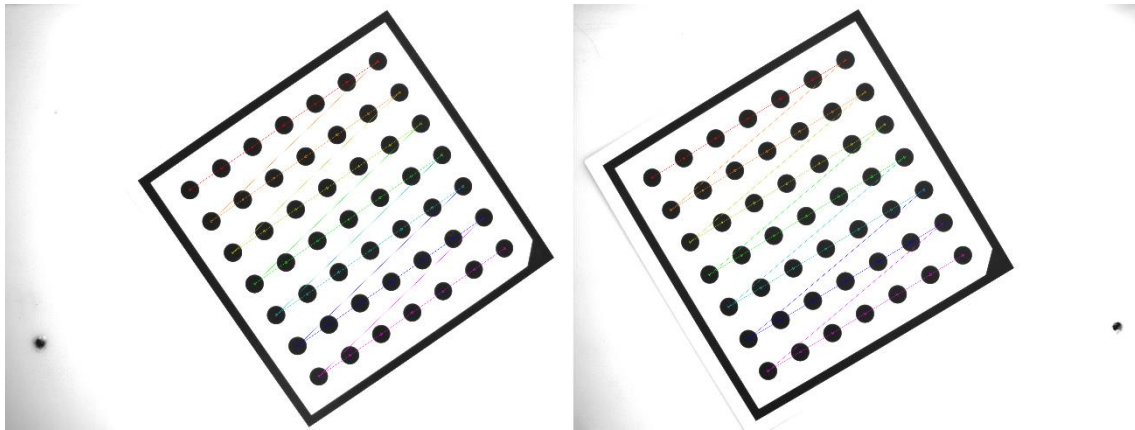


Figura 17: Ejemplo de visualización de la función *drawChessboardCorners*. En ambas imágenes se puede apreciar que todos los círculos fueron encontrados y que la correspondencia de los puntos es correcta.

Para terminar con este proceso previo al de calibración, debemos calcular los puntos del patrón en coordenadas del mundo real. Por suerte, el patrón de calibración es un elemento plano y rígido, por lo que podemos establecer que el componente Z de todos los puntos sea, por ejemplo, igual a cero. Además, cómo conocemos la distancia entre los centros (25mm, en este caso) y como éstos son equidistantes entre sí, podemos establecer el resto de coordenadas de manera sencilla. En el algoritmo implementado, los puntos se generan de la siguiente forma: $(25 * (\text{número_de_fila} - 1), 25 * (\text{número_de_columna} - 1), 0)$. Por tanto, las coordenadas del primer punto serán (0, 0, 0) y las del último (150, 150, 0). Puede parecer que, haciéndolo de esta manera, nos estemos inventando las coordenadas de los puntos del patrón de calibración. Y, en cierto modo, así es, ya que estamos definiendo los valores que más nos convienen. No obstante, ¿cuáles son las coordenadas correctas? ¿Cuál es el origen del sistema de coordenadas del mundo real? No lo sabemos. Podría ser cualquier punto del universo. Por eso mismo, podemos decidir el punto de origen que más nos convenga, que en nuestro caso es el centro del primer círculo del patrón.

Una vez calculados todos los puntos necesarios, podemos, al fin, hacer uso de la función *stereoCalibrate*. Esta función calculará, por un lado, los parámetros intrínsecos de las cámaras y los coeficientes de distorsión de las lentes y, por otro lado, la matriz de rotación y el vector de traslación entre la primera cámara y la segunda. Además, devolverá también un valor de salida, el error de reproyección (en píxeles). El error de reproyección es una medida que compara la proyección de un punto y las coordenadas reales de ese punto [29]. Sirve como indicador de la calidad de la calibración realizada. La función *stereoCalibrate* calibra las cámaras minimizando este error a lo largo de varias iteraciones. Resulta interesante reparar en el valor devuelto para saber la precisión de la calibración realizada, o para encontrar errores de implementación o funcionamiento (si el error devuelto es muy grande, 100 píxeles, por ejemplo, es posible que hayamos cometido algún error a la hora de calcular los puntos anteriormente comentados o a la hora de pasar los argumentos requeridos por *stereoCalibrate*).

4.4 Rectificación y corrección de la distorsión óptica

Una vez calibradas las cámaras y calculado los coeficientes de distorsión de las lentes, podemos utilizar la información obtenida para rectificar las imágenes y corregir la distorsión óptica.

En primer lugar, se han calculado, tanto para la primera cámara como para la segunda, las matrices de proyección y de rotación correspondientes al nuevo sistema de coordenadas rectificado. Para ello, se ha empleado la función *stereoRectify* [30], que toma los datos calculados en el paso anterior (matrices de parámetros intrínsecos de ambas cámaras, coeficientes de distorsión de las lentes, matriz de rotación y vector de traslación entre las cámaras) como parámetros de entrada y devuelve, como salida, las matrices de proyección y de rotación mencionadas.

Tras el cálculo de estas matrices, podemos aplicar una transformación a nuestras imágenes para que tengan la misma orientación y sean coplanares. Para ello, podemos utilizar la función *initUndistortRectifyMap* [31], que toma como parámetros de entrada la matriz de parámetros intrínsecos, los coeficientes de distorsión de la lente, la matriz de rotación rectificada y la matriz de proyección rectificada. Como resultado, la función generará un mapa de transformación horizontal y otro mapa de transformación vertical. Estos mapas pueden utilizarse en la función *remap* [32] para modificar las imágenes originales.

Sin embargo, pese a que es recomendable rectificar las imágenes y corregir la distorsión óptica antes de comenzar con el proceso de detección de puntos, según el caso, puede resultar más bien un inconveniente que una ayuda. Si entre ambas cámaras existen grandes diferencias de perspectiva, al aplicar el remapeado para que las imágenes sean coplanares y tengan la misma orientación, puede que parte de las imágenes rectificadas estén ocultas, además de aparecer zonas completamente negras o vacías. Esto se debe a que, al aplicar el cambio de perspectiva, puede ser que las coordenadas de las imágenes rectificadas no se correspondan con las coordenadas de las imágenes reales. Por tanto, es posible que haya zonas de las imágenes que no podamos analizar, o que las zonas negras virtualmente generadas alteren el funcionamiento normal de los algoritmos que más adelante se aplicarán. Para solucionar este problema, podríamos añadir una traslación adicional al mapeado anterior, que más adelante, en la triangulación de los puntos, deberá tenerse en cuenta para no afectar negativamente a la reconstrucción 3D. Todo esto puede ser realmente molesto, así que si nos encontramos en una situación similar, quizás sea mejor no rectificar las imágenes.

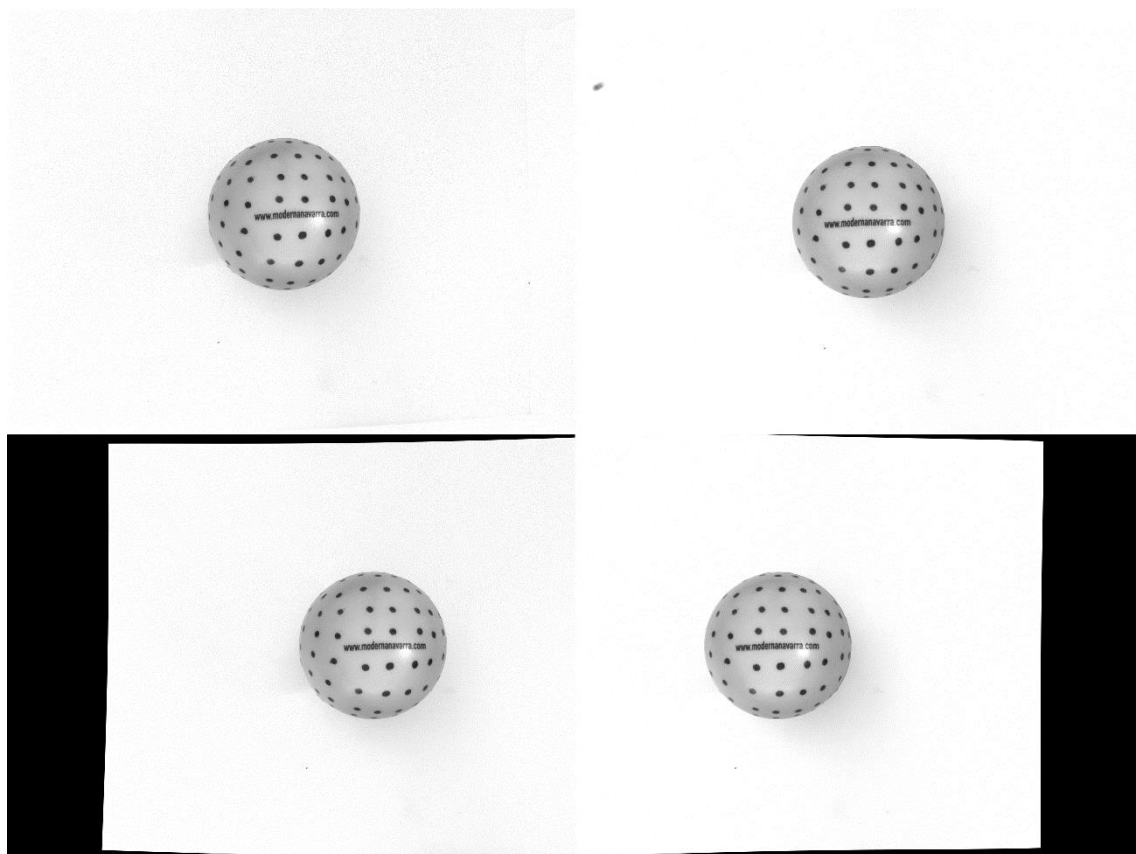


Figura 18: Arriba, par estéreo de las imágenes originales de la pelota. Abajo, el mismo par estéreo tras corregir la distorsión óptica y ser rectificado con *initUndistortRectifyMap* y *remap*. Al aplicar la transformación se puede apreciar que algunas zonas de las imágenes no son visbles.

Pero, ¿la rectificación y la corrección de la distorsión óptica no son pasos indispensables en la reconstrucción 3D? Sí, efectivamente, lo son. No obstante, la rectificación y la corrección de la distorsión óptica no tienen por qué hacerse antes de la detección de los puntos. Además, no necesitamos que se rectifique y se corrija la distorsión de toda la imagen. Realmente solo necesitamos que se aplique este proceso en los puntos que se vayan a reconstruir. Por tanto, primero podemos detectar los puntos a reconstruir y, después, rectificar y corregir la distorsión óptica de dichos puntos. Para ello, en vez de utilizar las funciones *initUndistortRectifyMap* y *remap*, podemos emplear la función *undistortPoints* [33]. Esta función toma los mismos parámetros de entrada que *initUndistortRectifyMap*, además de los puntos que hemos detectado, para devolvernos esos mismos puntos rectificados y sin distorsión óptica. En las pruebas realizadas se ha optado por utilizar esta alternativa, ya que genera los mismos resultados que la otra y, dada la relativamente baja cantidad de los puntos detectados es, además, más rápida.

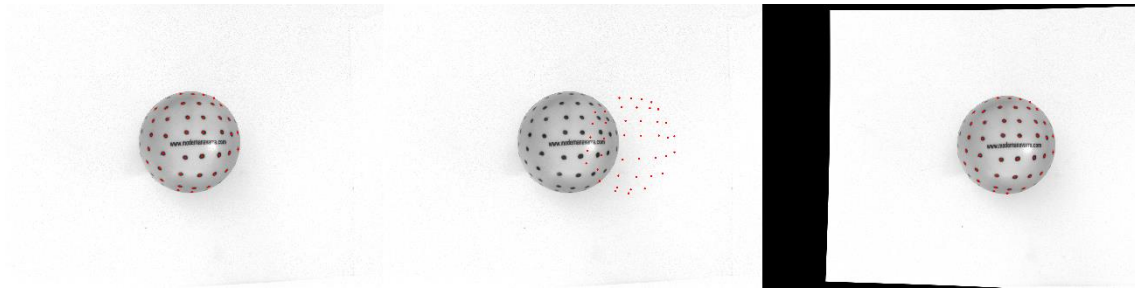


Figura 19: A la izquierda se muestran los puntos detectados en la imagen original. En el centro, esos mismos puntos tras corregir la distorsión óptica y ser rectificadas con la función *undistortPoints*. A la derecha, se muestra la imagen rectificada y corregida con *initUndistortRectifyMap* y *remap*, junto a los puntos rectificadas y corregidos con *undistortPoints*. Como se puede apreciar en la imagen, ambas formas de rectificar y corregir la distorsión óptica son equivalentes.

4.5 Detección de puntos

Como hemos visto en el apartado anterior, podemos posponer la rectificación y corrección de la distorsión óptica de las imágenes hasta que se hayan detectado los puntos deseados. Por ello, en las pruebas realizadas con los distintos objetos que se disponen, se ha decidido analizar las imágenes originales en vez de las rectificadas. En esta sección veremos las funciones y técnicas empleadas para la detección de los puntos a reconstruir de cada objeto.

4.5.1 Mangueta

Comencemos, pues, con la detección de los puntos de la mangueta. Para este objeto se han seleccionado un total de 23 puntos, de los cuales 7 de ellos corresponden a centros de agujeros o zonas circulares de la pieza, 15 pertenecen a distintas esquinas que tiene el objeto y, el punto restante, se corresponde con un punto negro sobre la mesa de trabajo.



Figura 20: Puntos detectados en la mangueta (imagen correspondiente a la cámara derecha).

En cuanto a la detección de los puntos circulares de la pieza, se ha empleado la función *HoughCircles* [34], que busca círculos en una imagen en escala de grises aplicando una modificación de la transformada de Hough [35]. Sin embargo, antes de aplicar el algoritmo y para facilitar la detección de los círculos, se ha aplicado un filtro de suavizado gaussiano a ambas imágenes con la función *GaussianBlur* [36]. Los parámetros empleados para el suavizado son una máscara de 5x5 y un valor de sigma (desviación estándar) de 2.

Tras aplicar el filtro gaussiano, se procede a detectar los círculos. La función *HoughCircles* toma varios parámetros de entrada. Entre los más relevantes se encuentran la imagen donde se buscarán los círculos, un primer umbral que se corresponde con el umbral más alto pasado al detector de Canny [37] (internamente, esta función utiliza un detector de bordes de Canny), un segundo umbral que será el umbral de acumulación de los centros de los círculos en la fase de detección (cuanto menor sea, mayor será la posibilidad de encontrar falsos círculos) y, para terminar, el radio mínimo y máximo de los círculos a detectar. Por tanto, para la detección del círculo central de la mangueta (punto N°14), se ha empleado un primer umbral de 50, un segundo umbral de 150 y un tamaño mínimo y máximo de radio de 250 y 300 píxeles, respectivamente. Para el resto de agujeros (puntos N°5, N°6, N°7, N°20 y N°23), en cambio, se ha utilizado un valor de 30 para el primer umbral y 100 para el segundo. El radio mínimo y máximo fijados, en este caso, han sido de 20 y 75 píxeles.

Tras detectar los centros de los agujeros y para terminar con este grupo de puntos, solo nos falta detectar el centro correspondiente a una zona redondeada de la pieza (punto N°19). Para facilitar la detección del mismo, se reduce el dominio de las imágenes mediante una ROI (*Region Of Interest*) [38] en la zona deseada, la cual se genera a partir del agujero más próximo previamente detectado (punto N°20). En concreto, la ROI es un rectángulo generado con la función *rectangle*, comenzando con la esquina superior izquierda en las coordenadas ($X - 400$ píxeles, $Y - 60$ píxeles) y terminando con la esquina inferior derecha en ($X + 25$ píxeles, $Y + 60$ píxeles), siendo X e Y las coordenadas del punto de referencia. Una vez determinada la nueva zona de análisis de la imagen, aplicamos, una vez más, la función *HoughCircles*. Los valores para esta vez son 15 para el primer umbral, 100 para el segundo, 5 píxeles como radio mínimo y 75 píxeles como radio máximo.

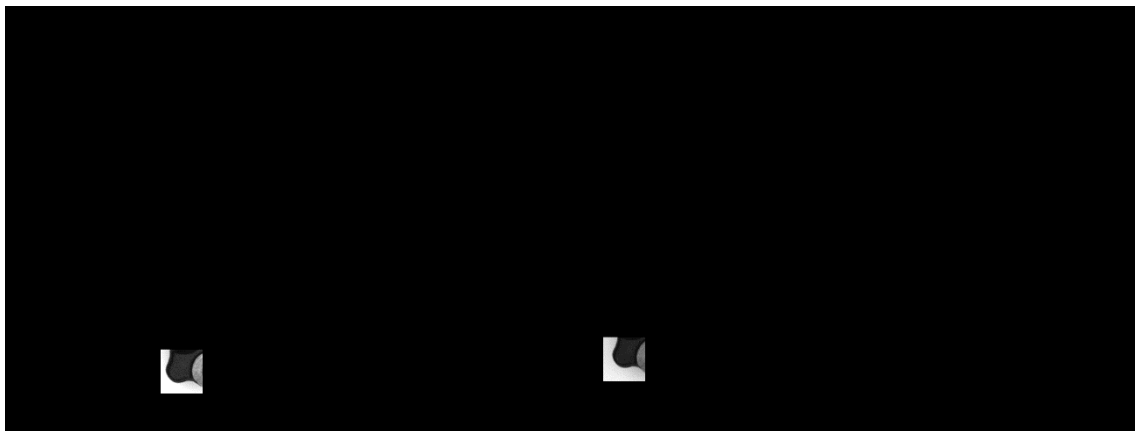


Figura 21: ROIs utilizadas para la detección del punto N°19.



Figura 22: Primer grupo de puntos detectados, correspondiente a zonas circulares de la mangueta.

Respecto a la detección de las esquinas de la mangueta, se ha empleado la función *cornerMinEigenVal* [39]. Esta función toma como parámetros de entrada, por un lado, la imagen donde se quieren detectar las esquinas y, por otro lado, el tamaño de la vecindad y el parámetro de apertura del algoritmo de Sobel [40] que utiliza internamente para

calcular la primera derivada (o una aproximación de ésta) de los píxeles. Como resultado, devuelve una imagen con los mínimos valores propios de la matriz de covarianza de las derivadas, siendo éstos valores números reales. Para poder analizar debidamente la imagen generada, se normalizan los valores entre 0 y 255 mediante la función *normalize* [41]. En todos los casos, los valores utilizados en esta función han sido un tamaño de 4 píxeles para la vecindad y una apertura de 3 píxeles para el algoritmo de Sobel.

No obstante, si se aplica la función *cornerMinEigenVal* a toda la imagen, la cantidad de esquinas detectadas puede ser inmensa. Si bien es cierto que, tras la normalización, podemos filtrar las esquinas generadas mediante umbrales, aun así, seleccionar los puntos deseados puede resultar laborioso. Por tanto, una vez más, para que la selección de los puntos deseados resulte más sencilla, se acota el análisis con distintas ROIs.

Con el fin de encontrar los puntos N°1, N°2, N°9, N°15, N°18 y N°22, se genera una ROI circular centrada en el agujero principal del objeto (punto N°14), con la ayuda de la función *circle* [42]. El radio del círculo generado es de 1090 píxeles para la imagen de la cámara izquierda y 950 píxeles para la imagen de la cámara derecha. A diferencia de la ROI rectangular generada previamente, no se reduce el dominio de la imagen al área del círculo, sino al revés, se acota la zona de análisis al área complementaria de éste. De esta forma, se limita la búsqueda a la periferia de la pieza y se eliminan una gran cantidad de esquinas que podrían generarse en el interior del objeto.

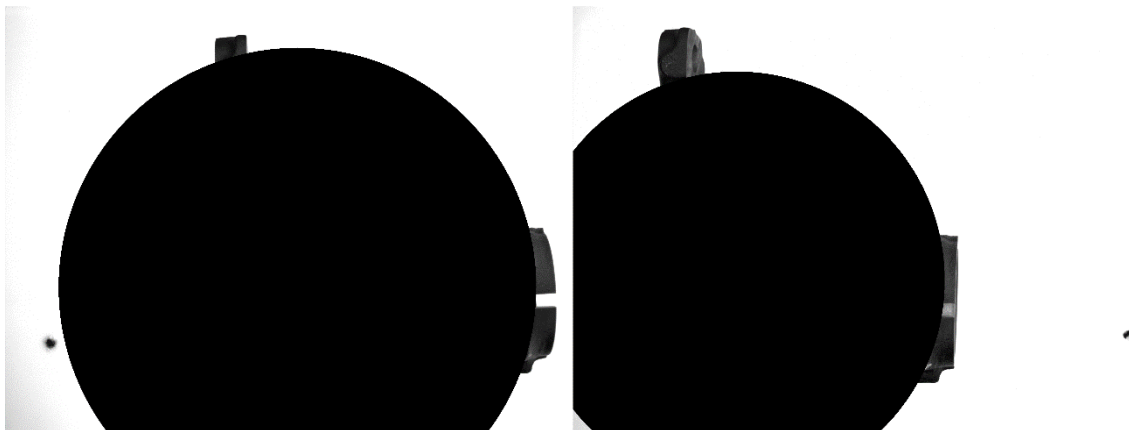


Figura 23: ROIs generadas para la detección de los puntos N°1, N°2, N°9, N°15, N°18 y N°22.

Una vez reducido el dominio de la imagen, se aplica la función *cornerMinEigenVal*, se normaliza la imagen de salida y se procede al filtrado de las esquinas encontradas. Para la selección del punto N°1, se utiliza un umbral inferior de 35 y un umbral superior de 70. Para el resto de esquinas, en cambio, el umbral inferior se aumenta a 50.



Figura 24: Puntos N°1, N°2, N°9, N°15, N°18 y N°22 detectados tras la umbralización.

En cuanto a detección de las esquinas restantes (puntos N°3, N°8, N°10, N°11, N°12, N°13, N°16, N°17 y N°21), se ha optado por seguir una estrategia similar a la llevada con los puntos anteriores. No obstante, en vez de generar una sola ROI en la que buscar todos los puntos, como están situados en zonas más conflictivas, se han generado varias regiones rectangulares con un área más reducida. En el Anexo 2 se puede observar tanto las coordenadas de la esquina superior izquierda como las coordenadas de la esquina inferior derecha de las ROIs rectangulares generadas para la detección de estos puntos, además de los umbrales utilizados para la selección de los mismos.

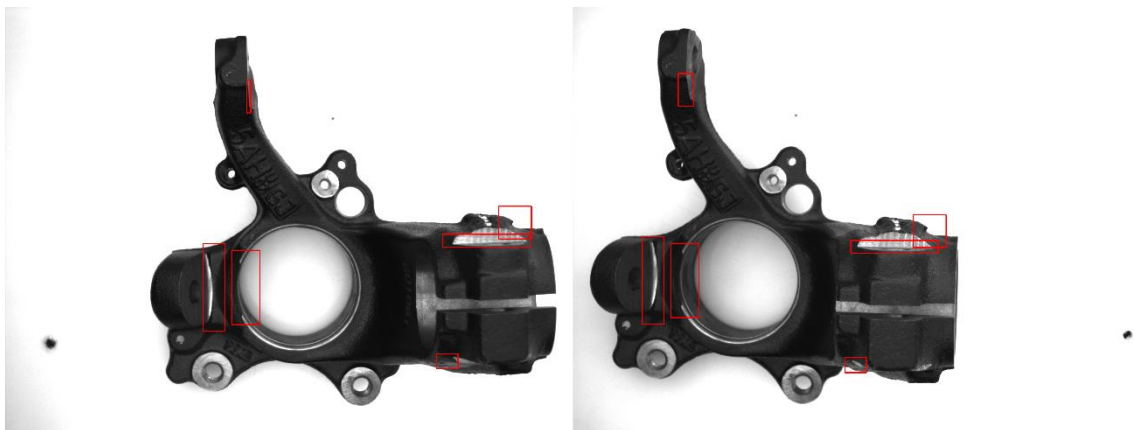


Figura 25: Conjunto de ROIs generadas para la detección de los puntos N°3, N°8, N°10, N°11, N°12, N°13, N°16, N°17 y N°21.

Por último, para detectar el punto oscuro sobre la mesa de trabajo, nuevamente se genera una ROI rectangular a partir del punto N°5, siendo $(X - 150, Y - 400)$ las coordenadas de la esquina superior izquierda y $(X + 150, Y - 100)$ las coordenadas de la esquina inferior derecha (la ROI se genera de la misma forma en ambas imágenes). No obstante, para este caso, la técnica empleada en la detección es diferente a las presentadas hasta ahora. Al tratarse de un punto tan pequeño, encontrarlo mediante la función *HoughCircles* puede resultar difícil. Por otro lado, *cornerMinEigenVal* busca esquinas y no puntos, por lo que

al utilizarla es probable que encuentre esquinas en el borde del punto, pero no su centro. Por tanto, en esta ocasión, para hallar el centro de este último punto, vamos a emplear un detector de “blobs”.



Figura 26: ROIs generadas para la detección del punto sobre la mesa de trabajo

Un “blob” es, simplemente, un conjunto de píxeles que comparten una serie de propiedades o características [43]. Es decir, es prácticamente lo mismo que una región o ROI de la imagen. A fin de cuentas, todos estos nombres vienen a representar lo mismo, una zona específica de la imagen que puede definirse mediante intensidades de gris, propiedades geométricas u otras características. Por tanto, la diferencia en la nomenclatura depende, básicamente, de nuestro propósito. Se suele hablar de “blobs” o regiones cuando vamos a analizar o filtrar una zona de la imagen y de ROIs cuando vamos a focalizar nuestro análisis a una zona específica de la imagen.

Una vez entendido esto, veamos cómo se ha construido el detector de “blobs” empleado. Primero, se instancia un objeto *SimpleBlobDetector::Params* [44] con los parámetros a emplear en la búsqueda del punto. En concreto, se activa la búsqueda por área y por color (se ponen a verdadero los parámetros *filterByArea* y *filterByColor*), se acota el área de los “blobs” seleccionados entre 0 y 1250 píxeles (con *minArea* y *maxArea*, respectivamente) y se indica que el color buscado es el negro (con *blobColor* igual a 0). Finalmente, se instancia el propio detector (objeto de la clase *SimpleBlobDetector*) con el conjunto de parámetros recién creados y se emplea la función *detect* del objeto instanciado. Como resultado, nos devolverá el punto que estamos buscando, terminando así con la detección del último punto de la mangueta.

4.5.2 Pelota

Tras ver cómo se han detectado los puntos a reconstruir de la mangueta, continuemos con el siguiente objeto: la pelota. Esta vez, el número total de puntos detectados son 42, que se corresponden con las manchas negras que ésta dispone sobre su superficie. Sin embargo, si reparamos al objeto, podemos observar que la cantidad de manchas es mayor. No obstante, pese a que las diferencias de perspectiva entre ambas cámaras no son muy grandes, algunas de estas manchas no son visibles (o apenas visibles) por las dos cámaras

al mismo tiempo. Por tanto, se ha decidido detectar solamente aquellas manchas presentes en las dos imágenes del par estéreo.

Con ese fin, como gran parte de la imagen no corresponde al objeto, se reduce el dominio de la imagen al círculo que forma la pelota. Para ello, primero, aplicamos de nuevo un filtro de suavizado gaussiano con la función `GaussianBlur`, utilizando los mismos parámetros que antes (máscara de 5x5 y desviación estándar de 2) y, después, empleamos la función `HoughCircles` con el propósito de encontrar tal círculo. Los valores utilizados esta vez son 55 para el primer umbral, 15 para el segundo, 300 píxeles como radio mínimo y 350 píxeles como radio máximo.

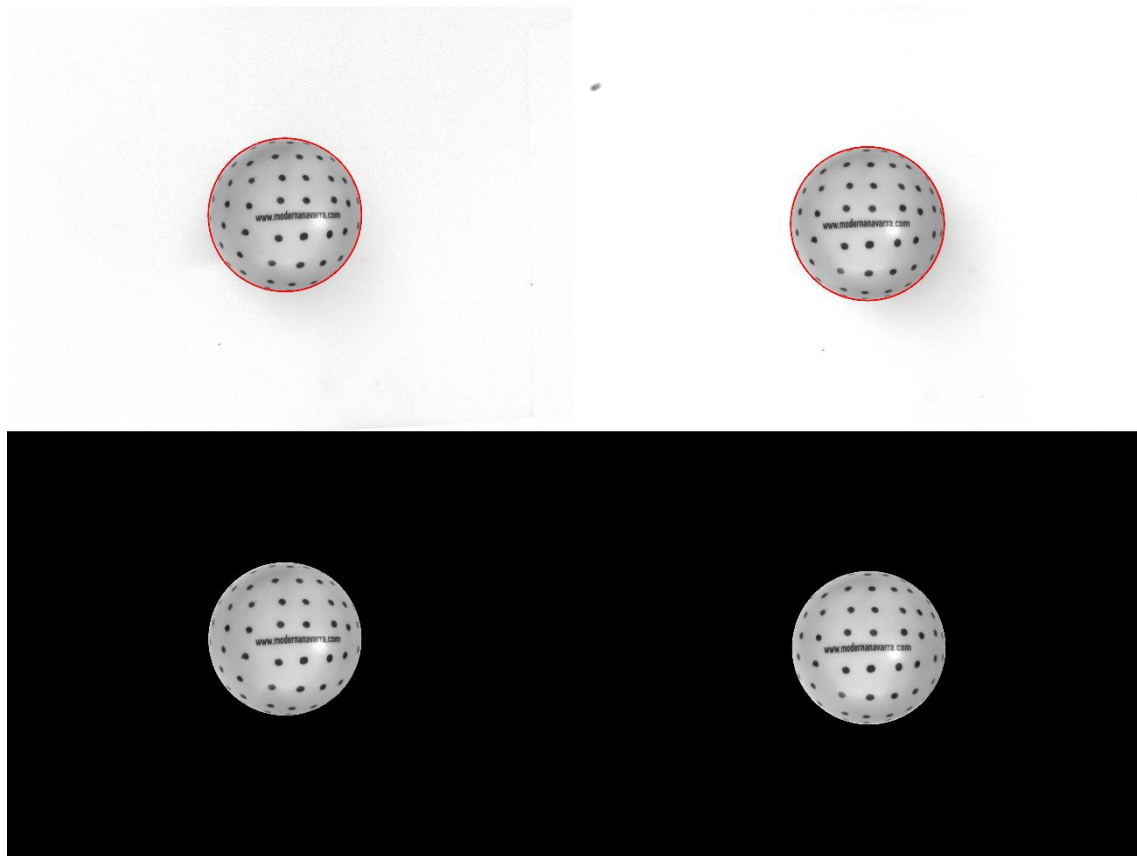


Figura 27: Círculo detectado por *HoughCircles* (arriba) y las nuevas imágenes tras reducir el dominio (abajo).

Una vez acotada la zona de análisis, se procede a detectar las manchas de la pelota. Esta vez, como las manchas no son círculos perfectos sino que son algo irregulares, en vez de la función *HoughCircles*, se vuelve a utilizar un detector de “blobs”. En esta ocasión, además de activar las mismas propiedades que la última vez (búsqueda por color y por área), se activa también la búsqueda por inercia y por convexidad (poniendo a verdadero los parámetros *filterByInertia* y *filterByConvexity*, respectivamente). Los rangos de valores utilizados para estas características son de 30 a 1500 píxeles para el área, de 0’01 a 1 para la inercia (parámetros *minInertiaRatio* y *maxInertiaRatio*) y de 0’92 a 1 para la

convexidad (parámetros *minConvexity* y *maxConvexity*). En cuanto al color, se indica que se desea buscar el color negro (parámetro *blobColor* igual a 0).

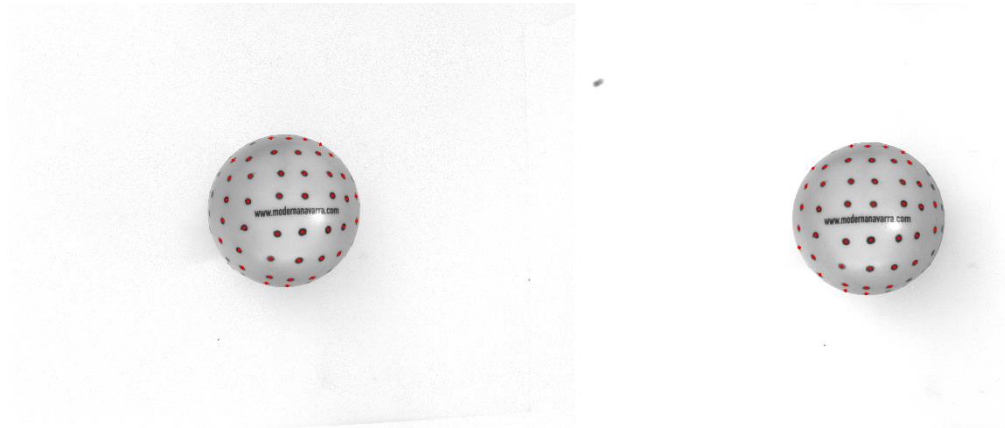


Figura 28: Puntos detectados tras aplicar el detector de “blobs”.

4.5.3 Cajas de cartón

Después de ver el proceso llevado a cabo para la detección de los puntos de la pelota, llega el momento de ver el tratamiento realizado a las imágenes de las cajas de cartón. Realmente, no se ha detectado ningún punto que pertenezca a las propias cajas. Se han colocado tres pequeños patrones de calibración, a distintas alturas, sobre la superficie de las cajas y se han detectado los círculos de éstos. Si bien los patrones, además de círculos, disponen de algunos anillos de color negro, éstos no han sido considerados para la reconstrucción tridimensional. Por tanto, el número de puntos a detectar en cada patrón es de 95, haciendo un total de 285.

Para ello, y con el objetivo de evitar la detección de puntos no deseados, en cada imagen se crean tres ROIs rectangulares que encapsulan los patrones de calibración. Las coordenadas de los rectángulos generados se pueden apreciar en las siguientes tablas:

ROI	Coordenadas esquina superior izquierda (píxeles)	Coordenadas esquina inferior derecha (píxeles)
ROI superior	(600, 200)	(1700, 770)
ROI central	(1330, 820)	(2070, 1250)
ROI inferior	(580, 1200)	(1300, 1650)

ROIs correspondientes a la imagen de la cámara izquierda

ROI	Coordenadas esquina superior izquierda (píxeles)	Coordenadas esquina inferior derecha (píxeles)
ROI superior	(670, 220)	(1620, 780)
ROI central	(1440, 860)	(2150, 1280)
ROI inferior	(600, 1200)	(1320, 1650)

ROIs correspondientes a la imagen de la cámara derecha

Tras reducir el área de análisis, se construye un detector de “blobs” para aplicarlo en cada región generada. En esta ocasión, se emplea la búsqueda por área y por color (negro), con un valor mínimo de área de 400 píxeles y un valor máximo de 800 píxeles.

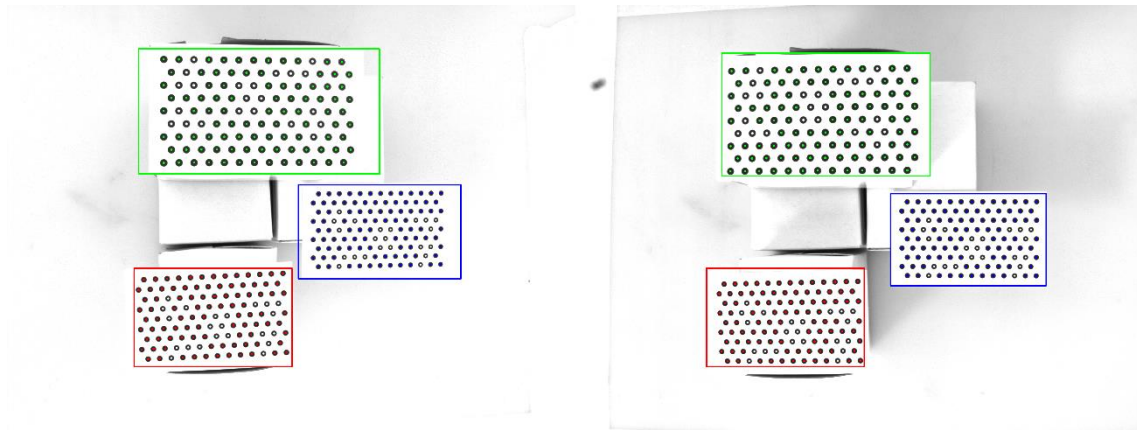


Figura 29: ROIs generadas y puntos detectados en el par estéreo de las cajas de cartón.

4.5.4 Escuadra metálica

Como último objeto contamos con una escuadra metálica. De forma análoga a las cajas de cartón, se han adherido dos patrones de calibración a las caras visibles de la pieza para detectar los círculos de éstos. Además, en este caso, también se han detectado los anillos del patrón que aparece en la parte superior de la imagen. Por lo que, esta vez, contamos con 117 puntos que provienen del primer patrón, junto con otros 95 puntos procedentes del segundo, haciendo un total de 212 puntos a detectar en este objeto.

La estrategia seguida para la escuadra es similar a la utilizada en las cajas de cartón. Primero se generan unas ROIs rectangulares entorno a los patrones, para luego aplicar un detector de “blobs” sobre los dominios reducidos. En las siguientes tablas se pueden observar las coordenadas de los rectángulos generados:

ROI	Coordenadas esquina superior izquierda (píxeles)	Coordenadas esquina inferior derecha (píxeles)
ROI superior	(830, 800)	(1500, 1020)
ROI inferior	(900, 1070)	(1400, 1730)

ROIs correspondientes a la imagen de la cámara izquierda

ROI	Coordenadas esquina superior izquierda (píxeles)	Coordenadas esquina inferior derecha (píxeles)
ROI superior	(970, 880)	(1620, 1060)
ROI inferior	(1040, 1100)	(1540, 1760)

ROIs correspondientes a la imagen de la cámara derecha

En cuanto a los parámetros empleados en el detector de “blobs”, son casi idénticos a los utilizados en el objeto anterior. Se vuelve a buscar por área y por color, pero esta vez, ya que por la perspectiva los círculos del patrón superior aparecen algo achatados, el rango del área filtrada aumenta ligeramente, siendo el valor mínimo de 100 píxeles y el máximo de 800 píxeles.



Figura 30: ROIs generadas y puntos detectados en el par estéreo de la escuadra metálica.

4.6 Correspondencia

Tras detectar todos los puntos deseados, llega el momento de resolver el problema de la correspondencia de los puntos. Si bien es cierto que OpenCV dispone de algoritmos para calcular la correspondencia de forma automática, como el *StereoBM* o *StereoSGBM*, dado que la cantidad de puntos detectados en los objetos analizados es relativamente baja, se ha decidido resolver este problema de forma semi-manual. Por tanto, primero, se han

empleado algoritmos de ordenación básicos para ordenar los puntos por filas o columnas, según el caso, para luego reordenar los puntos discrepantes de forma manual. Es cierto que, quizás, no sea ni la mejor forma ni la más elegante de hacerlo, pero recordemos que el objetivo del trabajo es medir la precisión que se puede lograr en la reconstrucción 3D con un sistema estéreo, por lo que, mientras el problema de la correspondencia se resuelva de forma satisfactoria, es irrelevante el medio o la técnica empleada.

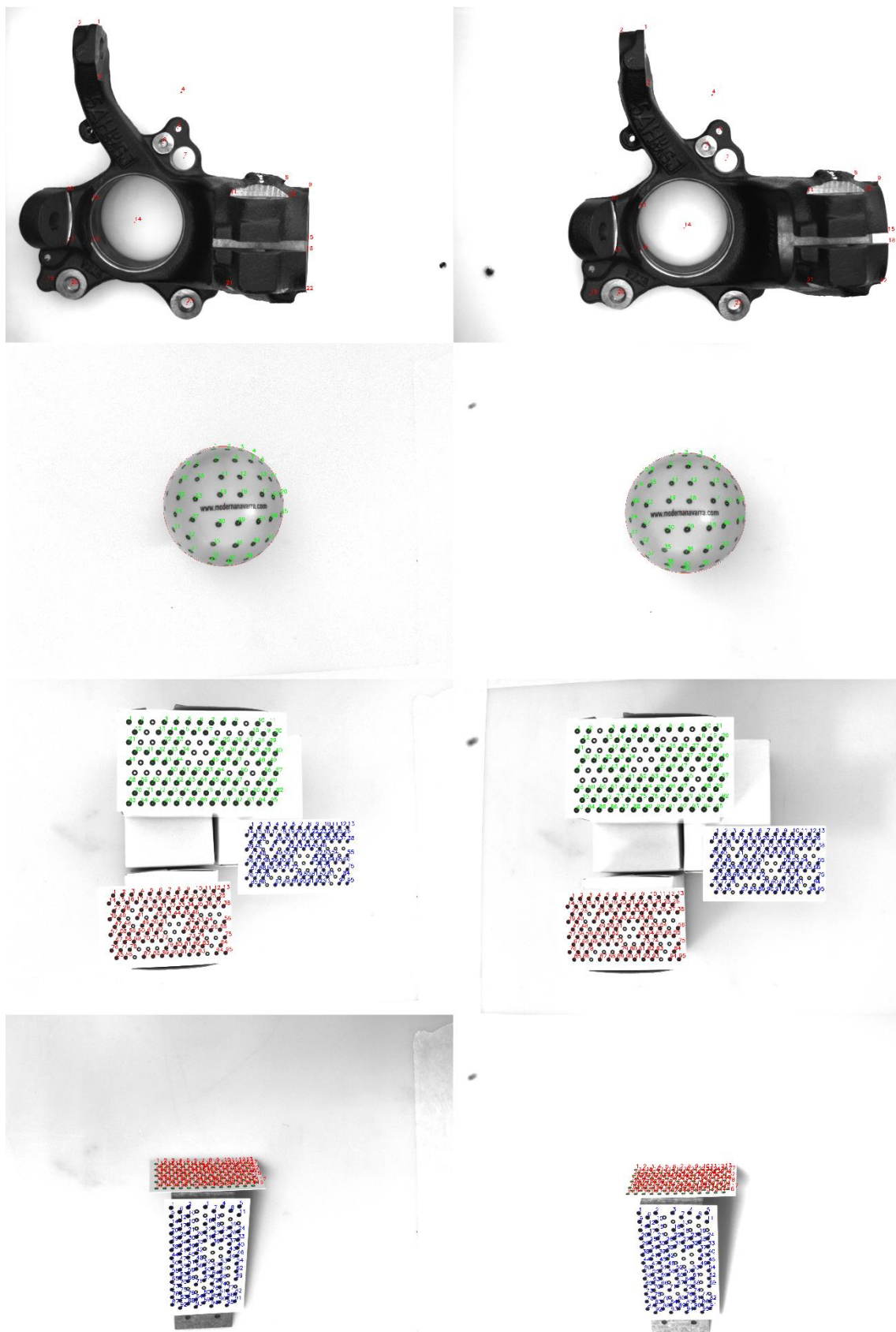


Figura 31: Correspondencia de puntos entre los pares estéreo de los cuatro objetos a reconstruir.

4.7 Triangulación

Tras resolver la correspondencia de los puntos, podemos realizar la reconstrucción 3D de éstos mediante triangulación. No obstante, antes de proceder con la triangulación, recordemos que el tratamiento llevado a cabo hasta el momento se ha efectuado sobre las imágenes originales. Por tanto, debemos rectificar los puntos y corregir la distorsión óptica de la lente con *undistortPoints*.

Una vez hecho esto, podemos utilizar la función *triangulatePoints* [45], que toma como parámetros de entrada, por un lado, las dos matrices de proyecciones rectificadas calculadas mediante *stereoRectify* y, por otro lado, los dos conjuntos de puntos detectados. Como resultado obtendremos una matriz de los puntos reconstruidos en coordenadas homogéneas, de dimensión $4 \times N$, donde N es el número de puntos reconstruidos. Por tanto, como último detalle, solamente nos falta pasar los puntos reconstruidos de coordenadas homogéneas a coordenadas del mundo real. Para ello, contamos con la función *convertPointsFromHomogeneous* [46], que se encarga de realizar esta conversión. Cabe mencionar que la matriz que toma como parámetro de entrada es de dimensión $N \times 4$, por lo que tendremos que trasponer la matriz de puntos reconstruidos para que la conversión pueda hacerse. Por suerte, OpenCV cuenta con la función *transpose* [47] para calcular la traspuesta de una matriz de forma sencilla.

5. Resultados

Una vez visto tanto el sistema binocular construido como el proceso llevado a cabo para reconstruir los distintos objetos presentados, reparemos a los resultados obtenidos. Cabe mencionar que, menos en el caso del patrón de calibración cuyas dimensiones y distancias son conocidas, el resto de medidas han sido tomadas con la ayuda de un calibre y un metro. Por tanto, los valores teóricos aquí presentados no están exentos de errores.

5.1 Calibración

Comencemos por analizar los resultados de las dos calibraciones realizadas. Por un lado tenemos la calibración del modelo estándar, en la que se han empleado 27 pares de imágenes del patrón de calibración y cuyo error de reproyección es de 0'6759 píxeles. Por otro lado, contamos con la calibración de la disposición de ejes ópticos cruzados, en la que se emplearon un total de 47 pares de imágenes y cuyo error de reproyección es de 1'2868 píxeles. Si bien se puede intuir una clara diferencia entre la precisión de ambas calibraciones, como las cámaras se colocaron a distintas alturas en cada disposición, la relación píxel-milímetro de sus correspondientes imágenes es diferente. Por tanto, vamos a calcular el error en milímetros reales, en vez de utilizar el obtenido en píxeles. Para ello, necesitamos averiguar la relación píxel-milímetro de cada disposición. Para esto podemos utilizar una herramienta online que nos proporciona Basler en su página web [48].

Seleccionamos el modelo de las cámaras utilizadas ("Basler ace acA2500-14gm") e introducimos tanto la distancia focal de las lentes (8mm) como la distancia de trabajo de las cámaras. La distancia de trabajo es algo menor a la distancia que hay de la mesa de trabajo al centro de sujeción de las cámaras, ya que se mide desde un punto en concreto de los objetivos. En nuestro caso, la distancia de trabajo para el primer modelo

implementado es de 534mm y 437,7mm para el segundo. La herramienta nos devolverá la resolución de la imagen en milímetros reales, siendo ésta de 375x281 milímetros para la primera configuración y 306x230 milímetros para la segunda. Como la relación píxel-milímetro varía ligeramente entre el eje horizontal y el vertical, utilizaremos el valor medio como medida, por lo que tendremos una relación de 6'9150 píxel/milímetro para el primer caso y 8'4614 píxel/milímetro para el segundo. Por tanto, obtenemos un error de proyección de 0'0978mm para la primera calibración y otro de 0,1521mm para la segunda.

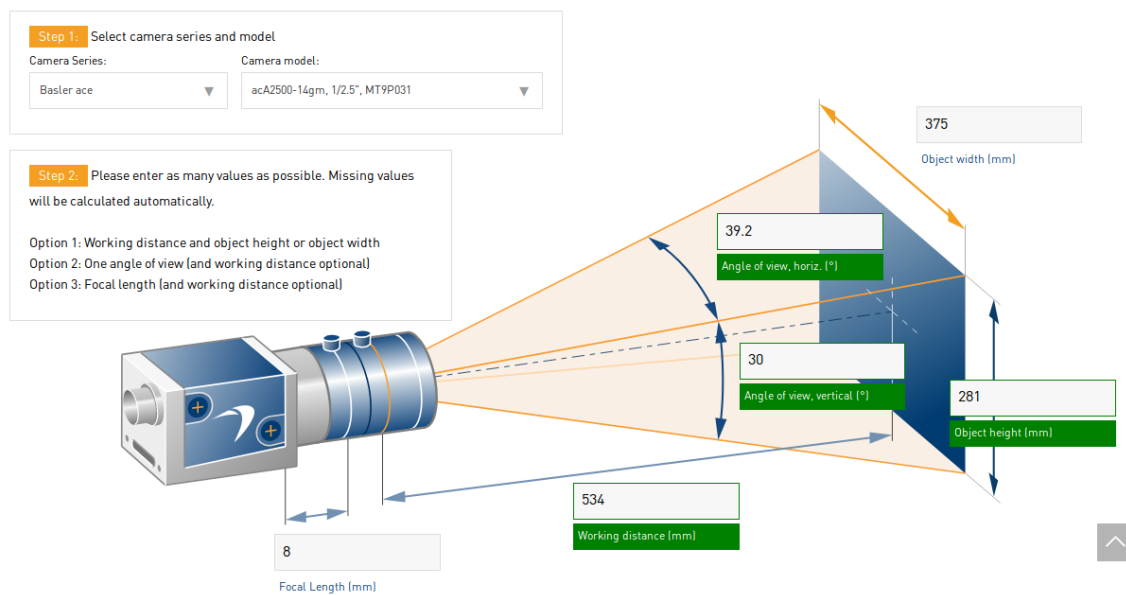


Figura 32: Herramienta de selección de cámaras y lentes de Basler [48]

Ahora que tenemos ambos errores en la misma escala, podemos analizar debidamente los resultados obtenidos en la calibración. Como hemos anticipado antes, existe una diferencia sustancial entre la primera calibración y la segunda. Esto puede deberse, principalmente, a dos razones. Por una parte, en la segunda calibración se tomaron 20 pares de imágenes más que en la primera, con una mayor variedad en la posición del patrón de calibración. Por tanto, no es extraño que se haya generado algo más de error por esta causa. Por otra parte, en el segundo caso existe una diferencia de orientación más acusada entre las cámaras (ejes ópticos cruzados), mientras que en el primero hay, prácticamente, un simple desplazamiento horizontal de una cámara a otra.

Todas estas causas han podido ser, en conjunto, la razón en la diferencia de los errores. No obstante, como veremos a continuación en los resultados obtenidos en la reconstrucción 3D de los objetos, el error de reproyección no es el único factor que afecta a la precisión de la propia reconstrucción.

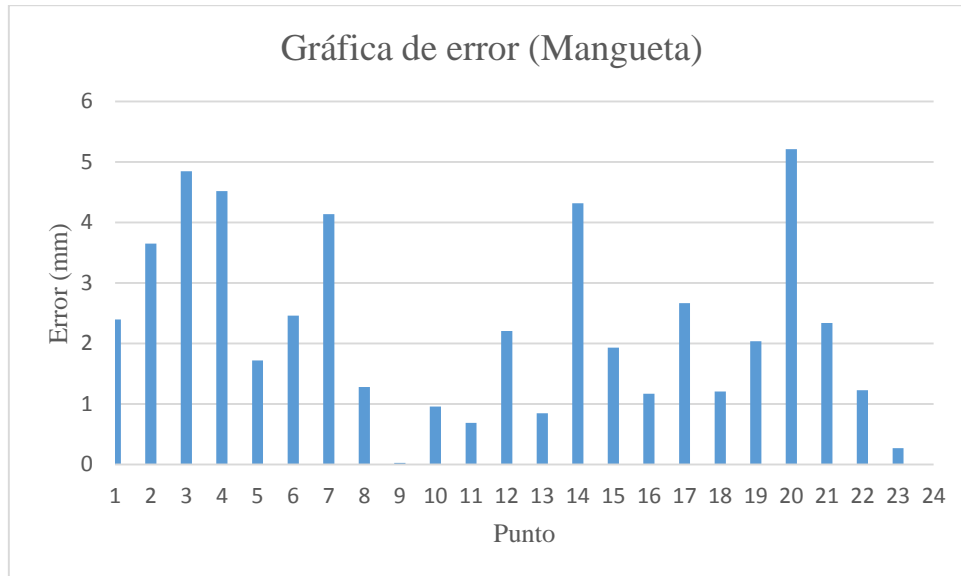
5.2 Mangueta

Respecto a la reconstrucción 3D realizada para los puntos detectados de la mangueta, en la siguiente tabla podemos apreciar los valores reales de la coordenada Z junto a los obtenidos mediante la triangulación, así como el error cometido para cada punto:

Punto	Reconstrucción (mm)	Valor real (mm)	Error (mm)
Nº1	104,4	102	2'4
Nº2	87,15	83,5	3'65
Nº3	95,85	91	4'85
Nº4	4,52	0	4'52
Nº5	42,28	44	1'72
Nº6	48,46	46	2'46
Nº7	39,86	44	4'14
Nº8	146,72	148	1,28
Nº9	149,03	149	0,03
Nº10	50,96	50	0,96
Nº11	160,31	161	0,69
Nº12	165,21	163	2,21
Nº13	54,85	54	0,85
Nº14	58,32	54	4,32
Nº15	175,57	177,5	1,93
Nº16	55,17	54	1,17
Nº17	47,33	50	2,67
Nº18	176,29	177,5	1,21
Nº19	26,96	29	2,04
Nº20	35,21	30	5,21
Nº21	156,66	159	2,34

N°22	146,77	148	1,23
N°23	30,27	30	0,27

Por tanto, el error medio cometido es de 2'2674mm y el máximo de 5,2100mm. En la siguiente gráfica se puede apreciar el error cometido para cada punto de una forma más visual.



Por último, en las siguientes imágenes se puede observar la reconstrucción 3D de los puntos seleccionados en la mangueta. El modelo tridimensional ha sido generado utilizando el programa de procesado y edición 3D conocido como Meshlab [49]. En concreto se ha utilizado la versión Meshlab 2016 para sistemas operativos Windows de 64 bits.

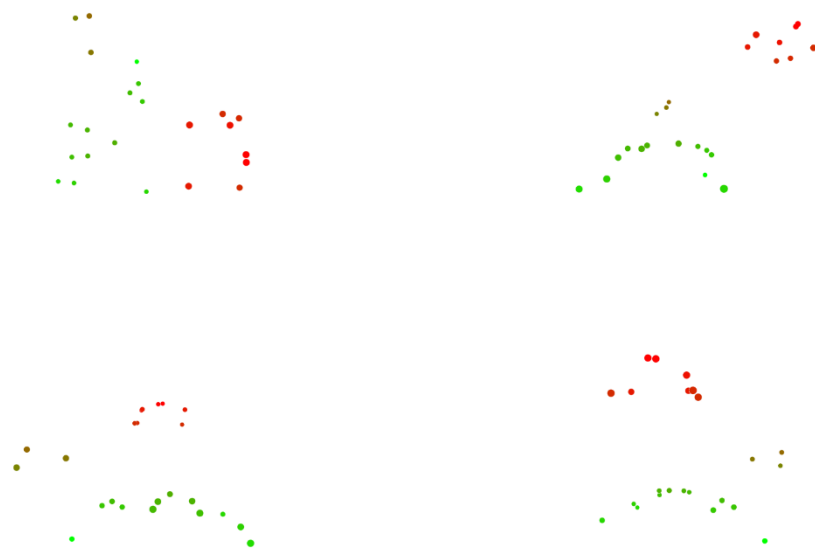
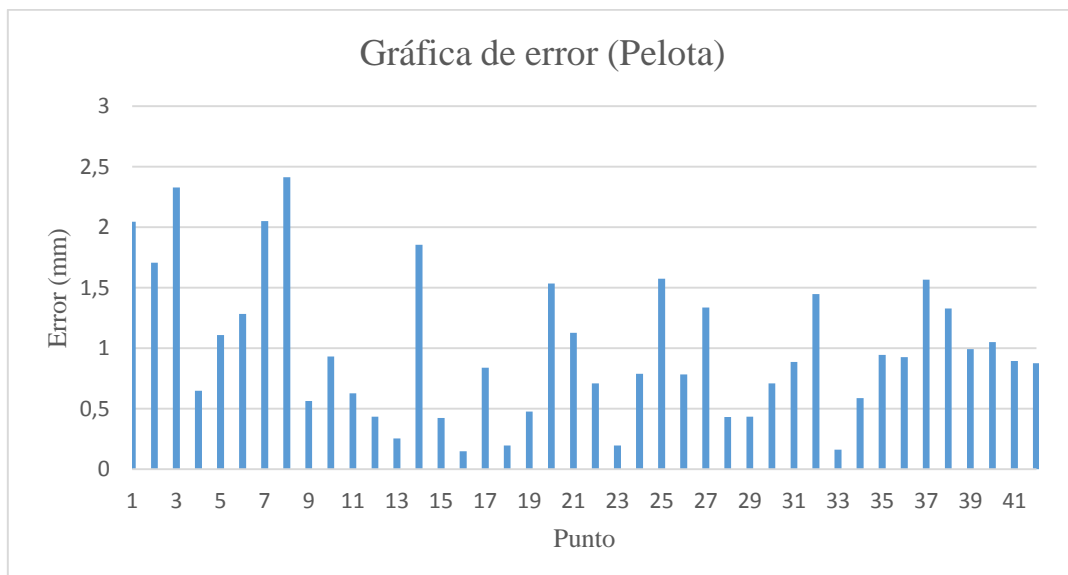


Figura 33: Modelo 3D de los puntos reconstruidos de la muñeca. Los puntos con tonos más verdosos indican menor altura. Los rojizos, en cambio, indican mayor altura.

5.3 Pelota

En cuanto a los resultados obtenidos en la reconstrucción de la pelota, el error máximo y medio cometido ha sido de 2'4115mm y 0'9911mm, respectivamente.



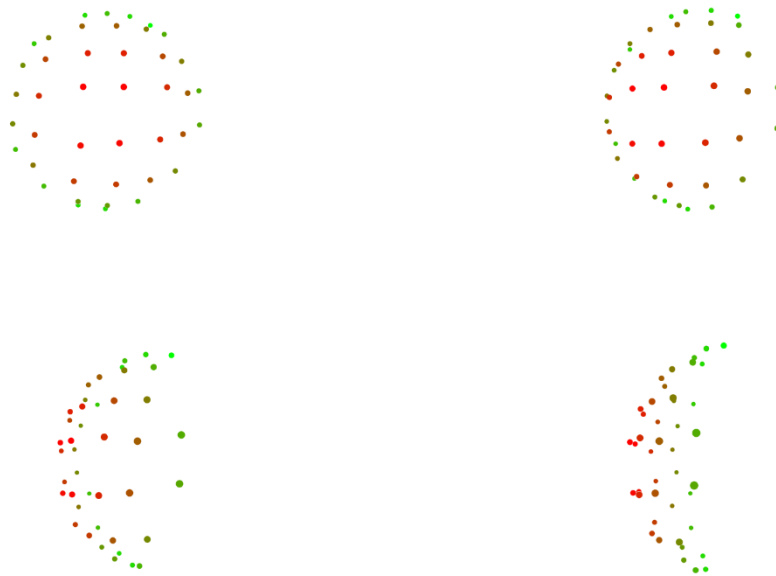
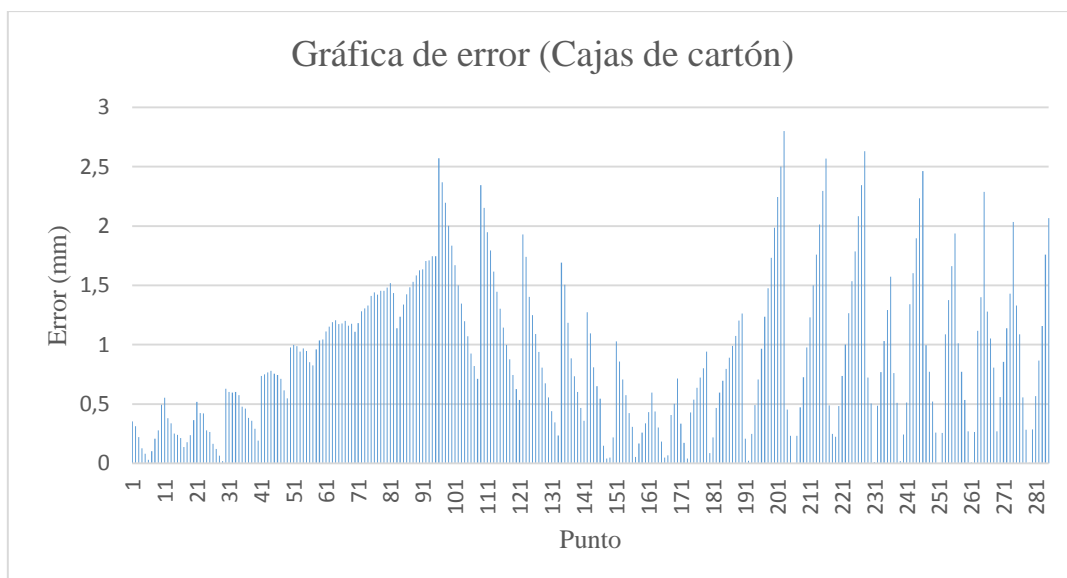


Figura 34: Modelo 3D de la pelota, creado con Meshlab. Los puntos con tonos más verdosos indican menor altura. Los rojizos, en cambio, indican mayor altura.

5.4 Cajas de cartón

Respecto a los puntos reconstruidos de las cajas de cartón, el error máximo cometido ha sido de 2'8005mm, mientras que el error medio ha sido de 0'9136mm.



Si observamos detenidamente el gráfico, podemos ver que el error cometido se comporta de forma “escalonada”. Esto se debe a que los puntos detectados pertenecen a unos patrones de calibración apoyados sobre unas cajas de cartón, cuya superficie no es

completamente plana y está algo abombada. Por tanto, es posible que los patrones de calibración no fuesen perfectamente paralelos a la mesa de trabajo. Al tomar las medidas de éstos, se asumió, por error, que cada patrón estaba a una única altura determinada. Por esa razón se comporta de esa forma el error generado. En las siguientes imágenes podemos ver el modelo 3D reconstruido con Meshlab.

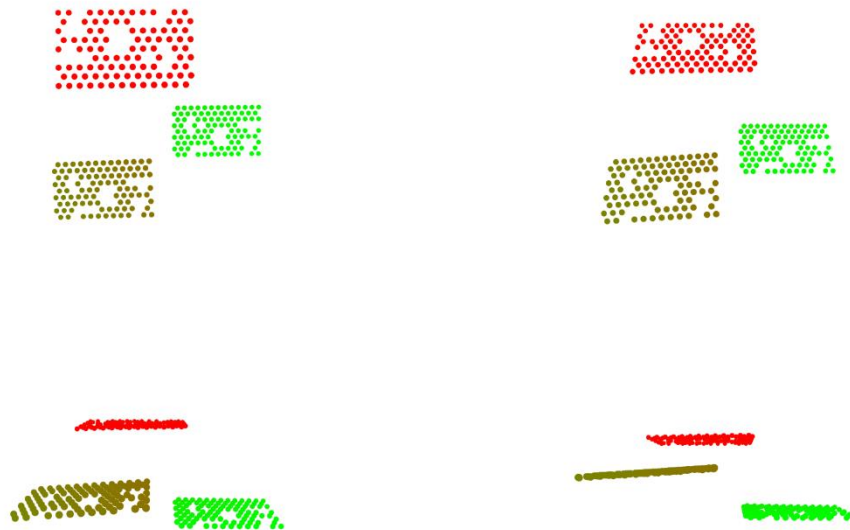
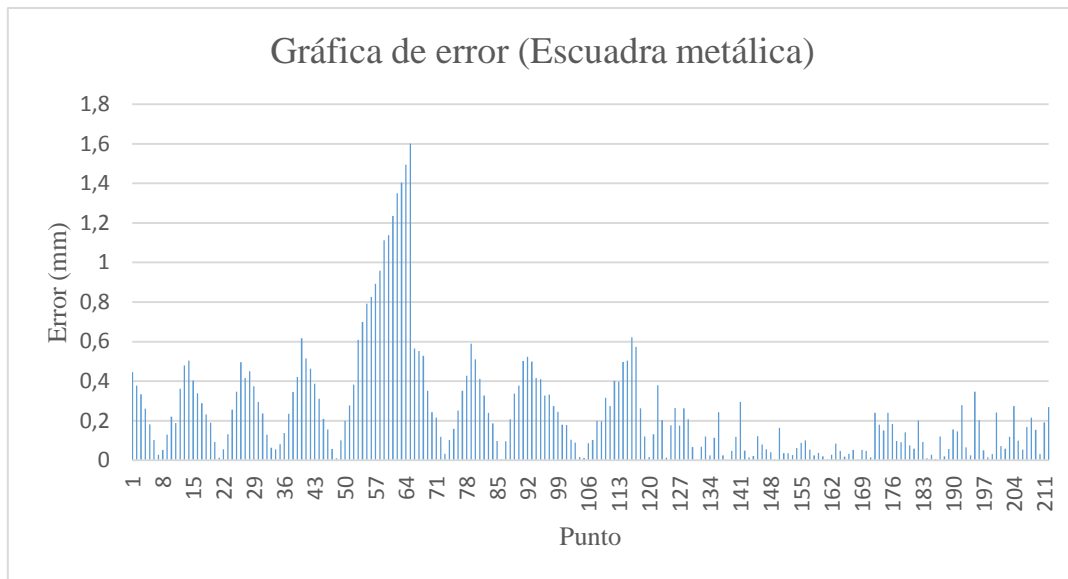


Figura 35: Modelo 3D de los patrones de calibración colocados sobre las cajas de cartón. Los puntos con tonos más verdosos indican menor altura. Los rojizos, en cambio, indican mayor altura

5.5 Escuadra metálica

Por último, los resultados obtenidos en la reconstrucción de la escuadra metálica muestran que se ha cometido un error máximo de 1'6006mm y un error medio de 0'2506mm.



Si reparamos con atención a la gráfica tenemos, de nuevo, un comportamiento del error similar al anterior. Aunque cabe decir que, esta vez, no es tan exagerado. En esta ocasión sucede por no haber adherido correctamente alineados los patrones de calibración a las caras de la escuadra metálica. A continuación tenemos, una vez más, el modelo 3D generado con Meshlab.

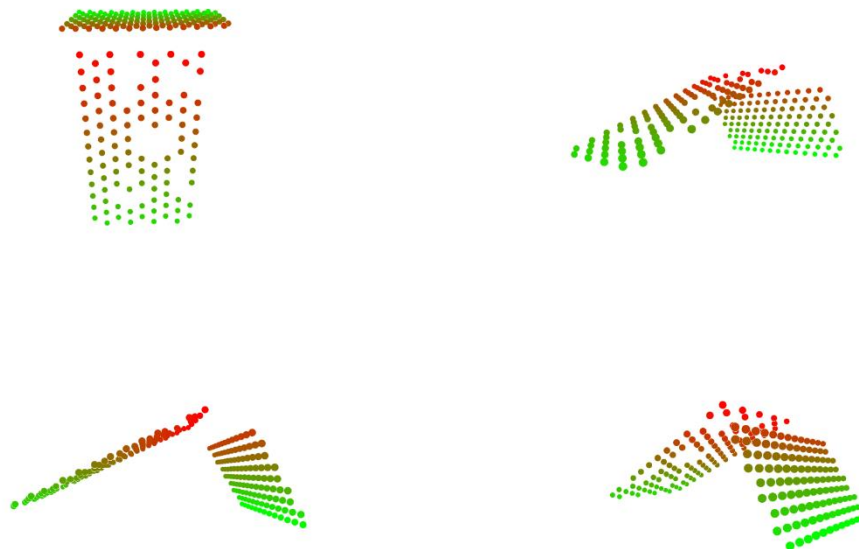


Figura 36: Modelo 3D de los patrones de calibración colocados sobre las caras de la escuadra metálica. Los puntos con tonos más verdosos indican menor altura. Los rojizos, en cambio, indican mayor altura.

6. Conclusiones

Una vez presentado los resultados obtenidos, veamos las conclusiones extraídas al analizar los mismos.

6.1 Conclusiones técnicas

En cuanto a la precisión alcanzada en la reconstrucción 3D de los objetos, podemos ver claramente que los objetos reconstruidos con el sistema binocular de ejes ópticos cruzados han logrado mejores resultados que el obtenido mediante el modelo estándar. Es cierto que, para extraer conclusiones más fiables, deberíamos hacer una cantidad mayor de pruebas. Sin embargo, con las cuatro reconstrucciones que hemos realizado, es suficiente para percatarnos de algunos detalles.

Para empezar, podemos ver que un menor error de reproyección no indica que, necesariamente, se vaya a lograr una mejor reconstrucción. Si bien es cierto que éste puede darnos una idea de la calidad de nuestra calibración, existen más factores además del error de reproyección que afectan a la reconstrucción 3D. Si nos fijamos en los resultados obtenidos, el error de reproyección de la primera calibración es menor al de la segunda. Sin embargo, los peores resultados han sido generados en la reconstrucción de la mangueta, que es precisamente el único objeto que ha sido reconstruido con el sistema correspondiente a la primera calibración.

Por tanto, debemos preguntarnos qué otros factores han podido perjudicar y beneficiar a los distintos sistemas construidos. Teniendo en cuenta que las técnicas empleadas de detección de puntos y correspondencia han sido similares en los objetos de ambos sistemas, la única razón subyacente es la propia disposición de las cámaras. Por una parte, en la estructura de ejes ópticos cruzados, al estar las cámaras más cerca de los objetos, la relación píxel-milímetro es mayor que en el primer caso, por lo que esto ha podido influir en la precisión de la reconstrucción. Por otra parte, el ángulo de intersección de los campos visuales de las cámaras del segundo sistema es mayor que el del primero, y como hemos visto en la sección 3.1.1, esto podía ser beneficioso para la reconstrucción 3D.

Por último, debemos reparar en la precisión conseguida, en general, con los sistemas binoculares construidos. Recordemos que uno de los objetivos principales de este trabajo es comprobar si un sistema estéreo puede llegar a competir con la precisión que proporciona una máquina de medición por coordenadas. Estas máquinas, en el peor de los casos, tienen una precisión de décima de milímetro. Sin embargo, la mejor reconstrucción que hemos realizado (escuadra metálica) tiene un error máximo de 1'6006mm. Por tanto, si bien no podemos afirmar rotundamente que lograr tal propósito sea imposible, es cierto que no se trata de una tarea trivial.

En conclusión, si bien hemos podido comparar el rendimiento de dos sistemas binoculares distintos y hemos visto cómo diferentes factores pueden afectar a la precisión de la reconstrucción 3D, con los modelos y técnicas implementadas no hemos podido alcanzar una calidad que pueda competir con las CMM.

6.2 Conclusiones personales

En cuanto a la experiencia obtenida a lo largo del desarrollo de este trabajo, he de decir que ha sido generalmente buena. Si bien me encuentro algo decepcionado con la precisión de los resultados obtenidos, ya que, al comienzo de este trabajo, pensé que con los sistemas binoculares implementados podría conseguir una precisión cercana a la décima de milímetro, he de admitir que tanto el proceso de investigación como de aprendizaje llevado a cabo a lo largo del mismo ha sido muy enriquecedor. Además, utilizar tecnologías (como OpenCV o Meshlab) que jamás había empleado, así como manipular las cámaras y sus objetivos, ha sido realmente entretenido. Personalmente, ha sido un proyecto desafiante para mí, por toda la cantidad de conceptos y herramientas que desconocía, pero he de admitir que me encuentro muy satisfecho por todo el trabajo realizado. Lo único que lamento es no haber tenido tiempo suficiente para realizar más pruebas y emplear otras técnicas o configuraciones de reconstrucción 3D.

6.3 Posibles mejoras

En cuanto a los posibles puntos a mejorar de este trabajo, podemos mencionar varias opciones que se podrían aplicar.

En primer lugar, se podría hacer un estudio exhaustivo de la calibración de los sistemas binoculares desarrollados. Como hemos podido observar, el error de reproyección de los distintos sistemas ha sido relativamente alto. Reducir el error en este apartado podría mejorar sustancialmente la calidad de las reconstrucciones tridimensionales realizadas.

En segundo lugar, sería aconsejable reconstruir piezas que hayan sido debidamente validadas y de las que se disponga un plano o documento de sus dimensiones. Al fin y al cabo, los valores teóricos de las piezas reconstruidas para este trabajo han sido tomadas mediante un metro y un calibre, por lo que los errores generados han sido calculados frente a valores que podrían ser diferentes a los reales.

Por último, cabe mencionar que las cámaras y objetivos empleados no tienen por qué ser los más adecuados para los objetos reconstruidos. Si se quisiera mejorar la precisión de los objetos reconstruidos, se debería realizar un profundo estudio sobre las características de las cámaras y lentes del mercado, para poder seleccionar aquellas que mejor se adapten a nuestro caso.

6.4 Líneas futuras

Respecto a futuros trabajos que se podrían realizar a partir de este, además de implementar las mejoras propuestas en el apartado anterior, existe una gran cantidad de líneas a seguir.

Por una parte, se podría aumentar el número de cámaras utilizado, para analizar la precisión conseguida mediante un sistema estéreo multicámara. Al aumentar el número de proyecciones de un punto de la escena, se puede reducir el error cometido en la reconstrucción tridimensional, y extender un sistema binocular a uno multicámara no implicaría grandes cambios en los conceptos presentados en este trabajo [50].

Por otra parte, se podría realizar una reconstrucción 3D densa. En este caso, el proceso de calibración y el de triangulación serían los mismos, pero habría que adaptar al nuevo

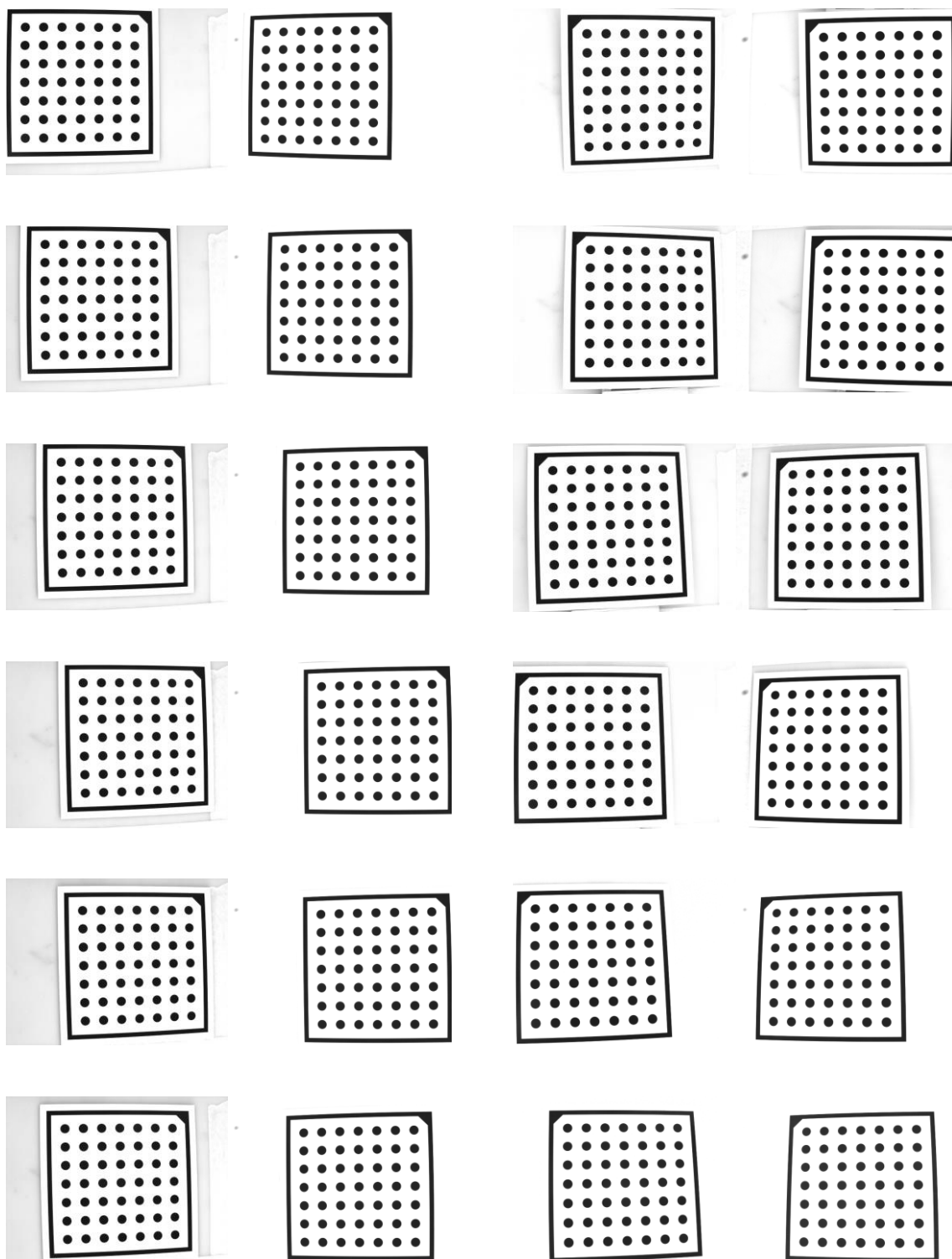
tipo de reconstrucción las técnicas empleadas de detección de puntos y resolución de la correspondencia (por ejemplo, aplicar algoritmos de detección de texturas, utilizar algoritmos de resolución de correspondencia automáticos que minimizan el error cometido, generar mapas de profundidad o disparidad...).

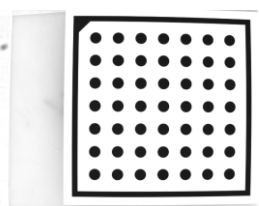
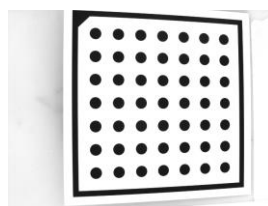
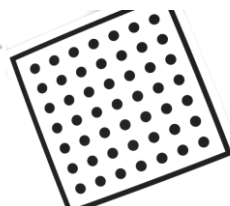
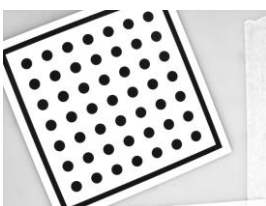
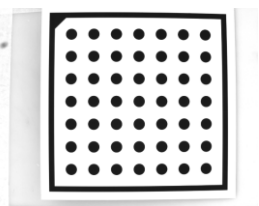
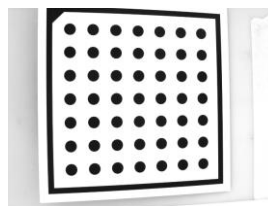
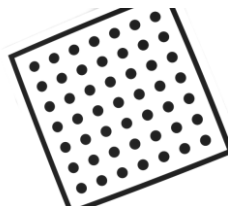
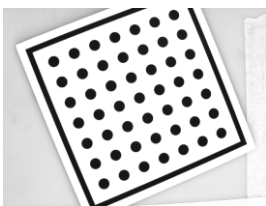
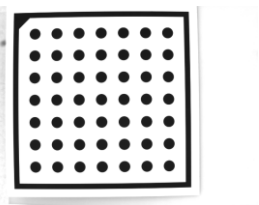
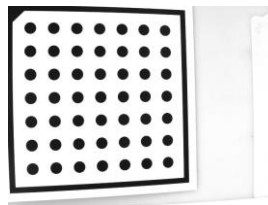
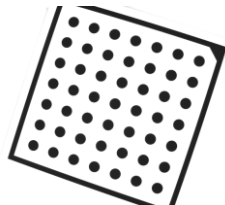
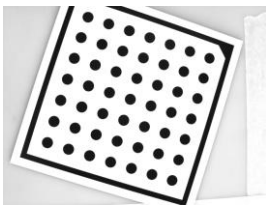
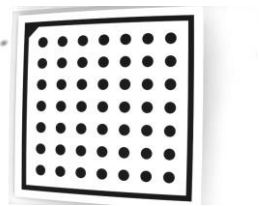
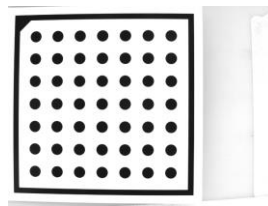
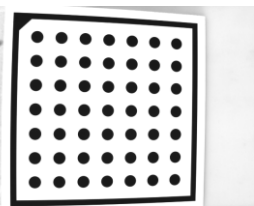
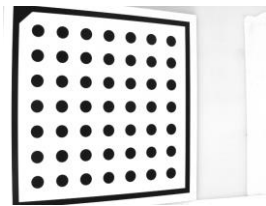
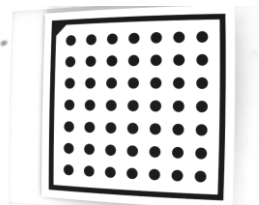
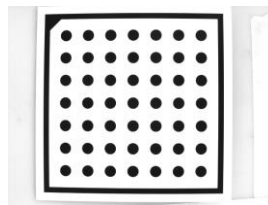
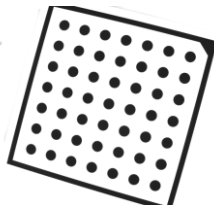
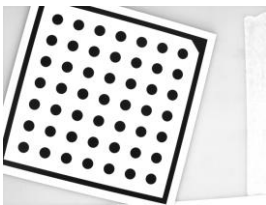
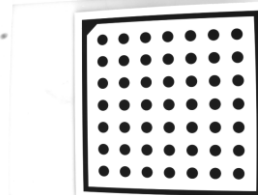
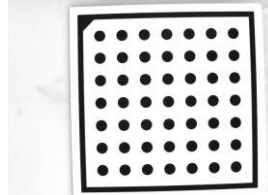
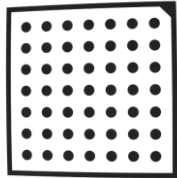
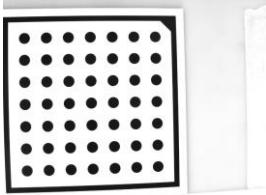
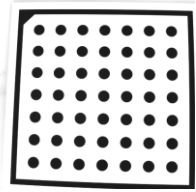
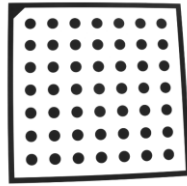
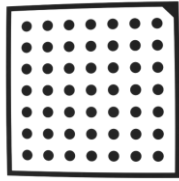
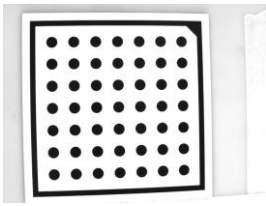
Los conceptos y técnicas presentadas en este trabajo son la base de la reconstrucción tridimensional con sistemas estéreo, por lo que, como podemos ver, la cantidad de trabajos futuros que se podrían hacer siguiendo esta misma línea son innumerables.

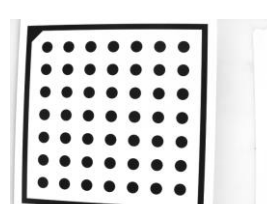
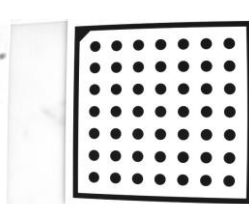
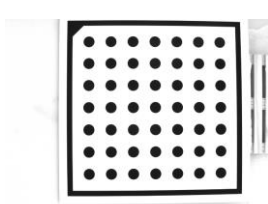
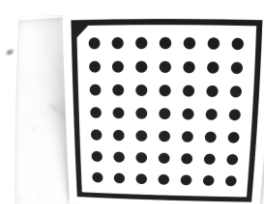
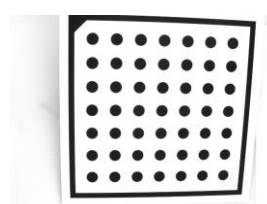
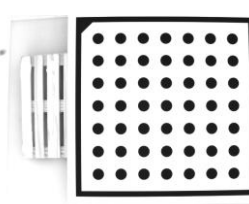
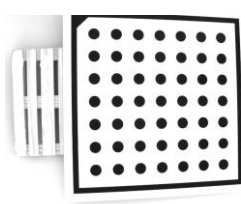
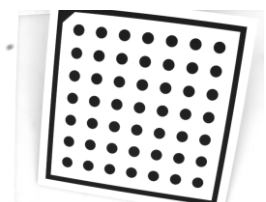
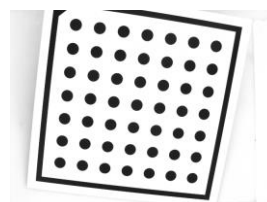
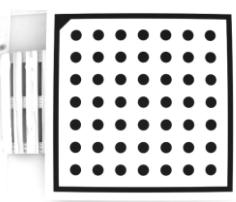
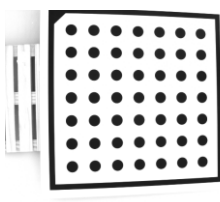
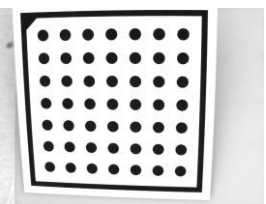
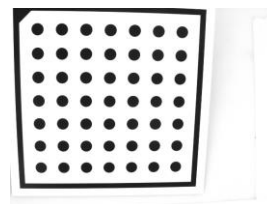
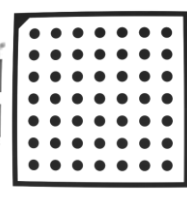
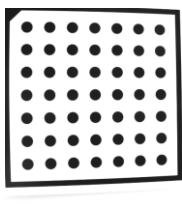
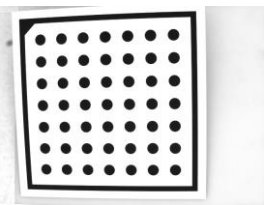
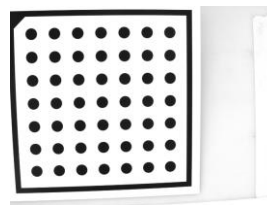
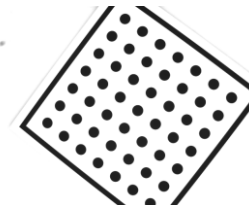
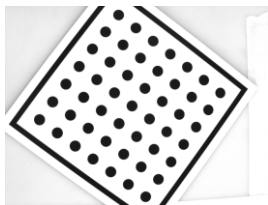
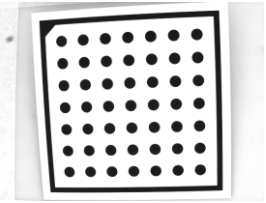
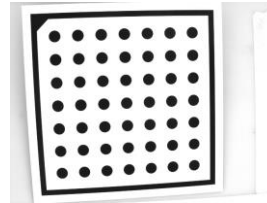
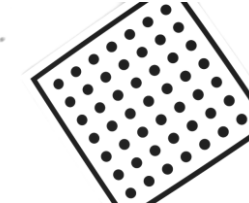
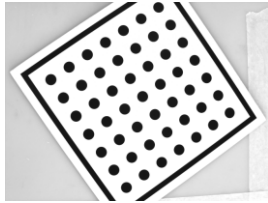
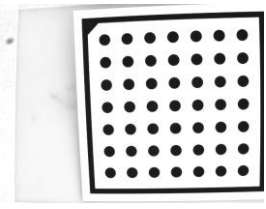
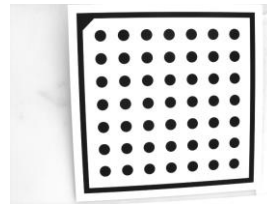
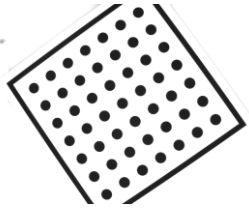
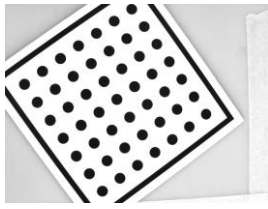
7. Anexos

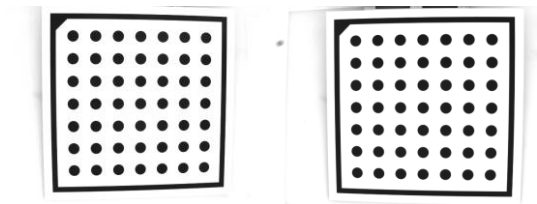
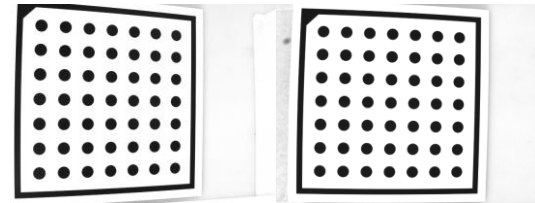
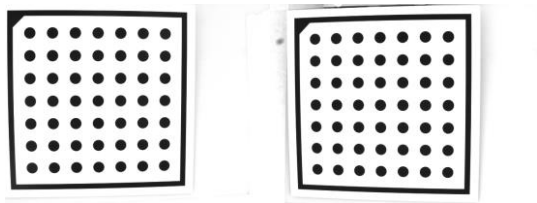
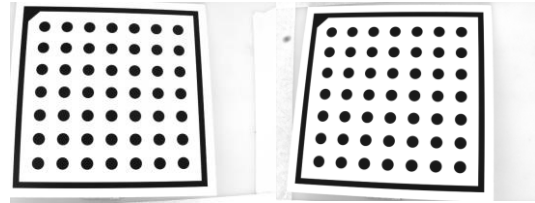
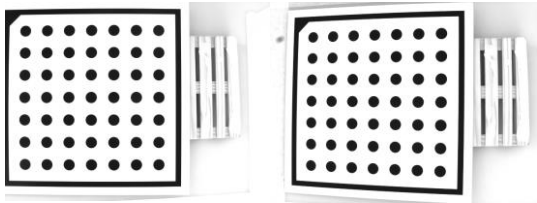
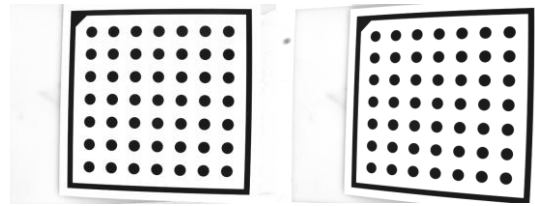
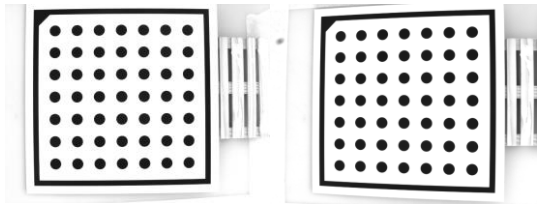
Anexo I

47 pares estéreo del patrón de calibración para la configuración del sistema binocular con ejes ópticos cruzados.









Anexo II:

Tablas que muestran las coordenadas de las ROIs rectangulares empleadas en la detección de los puntos N°3, N°8, N°10, N°11, N°12, N°13, N°16, N°17 y N°21, así como los umbrales utilizados en la selección de los mismos. X e Y se corresponden con las coordenadas del **punto de referencia**.

Punto a detectar	Punto de referencia	Coordenadas esquina superior izquierda (píxeles)	Coordenadas esquina inferior derecha (píxeles)
N°3	N°1	$(X - 50, Y + 200)$	$(X + 20, Y + 350)$
N°8	N°9	$(X - 200, Y - 100)$	$(X - 50, Y + 20)$
N°10	N°14	$(X - 430, Y - 200)$	$(X - 330, Y + 200)$
N°11	N°8	$(X - 350, Y + 60)$	$(X + 50, Y + 120)$
N°12	N°8	$(X - 350, Y + 60)$	$(X + 50, Y + 120)$
N°13	N°14	$(X - 300, Y - 170)$	$(X - 170, Y + 170)$
N°16	N°14	$(X - 300, Y - 170)$	$(X - 170, Y + 170)$
N°17	N°14	$(X - 430, Y - 200)$	$(X - 330, Y + 200)$
N°21	N°22	$(X - 500, Y - 50)$	$(X - 400, Y + 20)$

Tabla correspondiente a la imagen de la cámara izquierda

Punto a detectar	Punto de referencia	Coordenadas esquina superior izquierda (píxeles)	Coordenadas esquina inferior derecha (píxeles)
Nº3	Nº1	(X, Y + 200)	(X + 20, Y + 350)
Nº8	Nº9	(X - 200, Y - 100)	(X - 50, Y + 20)
Nº10	Nº14	(X - 430, Y - 200)	(X - 330, Y + 200)
Nº11	Nº8	(X - 320, Y + 60)	(X + 80, Y + 120)
Nº12	Nº8	(X - 320, Y + 60)	(X + 80, Y + 120)
Nº13	Nº14	(X - 300, Y - 170)	(X - 170, Y + 170)
Nº16	Nº14	(X - 300, Y - 170)	(X - 170, Y + 170)
Nº17	Nº14	(X - 430, Y - 200)	(X - 330, Y + 200)
Nº21	Nº22	(X - 500, Y - 20)	(X - 400, Y + 50)

Tabla correspondiente a la imagen de la cámara derecha

Punto a detectar	Umbral inferior	Umbral superior
Nº3	100	125
Nº8	50	70
Nº10	10	125
Nº11	40	125
Nº12	40	125
Nº13	110	125
Nº16	110	125
Nº17	10	125
Nº21	115	125

Tabla que muestra los umbrales utilizados, en ambas imágenes, para la selección de los **puntos a detectar**

8. Bibliografía

- [1] Telescopio HiRISE. [En línea]. Disponible en: <https://mars.nasa.gov/mro/mission/instruments/hirise/>
- [2] Reconstrucción 3D. [En línea]. Disponible en: https://en.wikipedia.org/wiki/3D_reconstruction
- [3] Máquina de medición por coordenadas. [En línea]. Disponible en: <https://goo.gl/1sX1PY>
- [4] Geert De Cubber, Lazaros Nalpantidis, Georgios Ch. Sirakoulis y Antonios Gasteratos, “Intelligent Robots need Intelligent Vision: Visual 3D Perception”. [En línea]. Disponible en: <https://goo.gl/zpgoWT>
- [5] Arnadi D. Murtiyoso y Deni Suwardhi, “A Comparison of Sparse and Dense Point Approach to Photogrammetric 3D Modeling for Stone Textured Objects (Case Study: Archeological Sites)”. [En línea]. Disponible en: <https://goo.gl/2q6Nww>
- [6] Cámara estenopeica. [En línea]. Disponible en: <https://goo.gl/BQNEap>
- [7] Modelo de la cámara estenopeica. [En línea]. Disponible en: https://en.wikipedia.org/wiki/Pinhole_camera_model
- [8] Modelo de la cámara estenopeica. [En línea]. Disponible en: <https://www.mathworks.com/help/vision/ug/camera-calibration.html>
- [9] Parámetros intrínsecos y extrínsecos. [En línea]. Disponible en: <https://goo.gl/wEuTaK>
- [10] Distorsión óptica. [En línea]. Disponible en: [https://en.wikipedia.org/wiki/Distortion_\(optics\)](https://en.wikipedia.org/wiki/Distortion_(optics))
- [11] Jianhua Wang, Fanhuai Shi, Jing Zhang y Yuncai Liu, “A new calibration model of camera lens distortion”. doi: 10.1016/j.patcog.2007.06.012
- [12] Julie Delon y Bernard Rougé, “Small Baseline Stereovision”. [En línea]. Disponible en: <https://goo.gl/eJjxrL>
- [13] Modelo estándar. [En línea]. Disponible en: <http://www.dis.uniroma1.it/~iocchi/stereo/triang.html>
- [14] Fotogrametría. [En línea]. Disponible en: <https://goo.gl/uDHmPX>
- [15] Zhengyou Zhang, “A Flexible New Technique for Camera Calibration”. [En línea]. Disponible en: <https://goo.gl/sWcf2D>
- [16] Barbara Frank, Cyrill Stachniss, Giorgio Grisetti, Kai Arras y Wolfram Burgard, “Robotics 2 Camera Calibration”. [En línea]. Disponible en: <https://goo.gl/HqTcj4>

- [17] Rectificación. [En línea]. Disponible en:
https://en.wikipedia.org/wiki/Image_rectification
- [18] Daniel Oram, “Rectification for Any Epipolar Geometry”. [En línea]. Disponible en: <https://goo.gl/RJiWTF>
- [19] Correspondencia. [En línea]. Disponible en:
https://en.wikipedia.org/wiki/Correspondence_problem
- [20] Myron Z. Brown, Darius Burschka y Gregory D. Hager, “Advances in Computational Stereo”. [En línea]. Disponible en: <https://goo.gl/QrBX14>
- [21] Ning Quian, “Binocular Disparity and the Perception of Depth”. [En línea]. Disponible en: <https://goo.gl/pvj4Dr>
- [22] “Stereo and 3D Vision”. [En línea]. Disponible en: <https://goo.gl/fesKoF>
- [23] GigE Vision. [En línea]. Disponible en: <https://goo.gl/fZuTYm>
- [24] OpenCV. [En línea]. Disponible en: <https://opencv.org/>
- [25] OpenCV, *stereoCalibrate*. [En línea]. Disponible en: <https://goo.gl/2Ddk17>
- [26] Sourish Ghosh, “Stereo calibration using C++ and OpenCV”. [En línea]. Disponible en: <https://sourishghosh.com/2016/stereo-calibration-cpp-opencv/>
- [27] OpenCV, *findCirclesGrid*. [En línea]. Disponible en: <https://goo.gl/WXbDvn>
- [28] OpenCV, *drawChessboardCorners*. [En línea]. Disponible en:
<https://goo.gl/rXTJqq>
- [29] Error de reproyección. [En línea]. Disponible en:
https://en.wikipedia.org/wiki/Reprojection_error
- [30] OpenCV, *stereoRectify*. [En línea]. Disponible en: <https://goo.gl/zNaQCt>
- [31] OpenCV, *initUndistortRectifyMap*. [En línea]. Disponible en:
<https://goo.gl/fo72of>
- [32] OpenCV, *remap*. [En línea]. Disponible en: <https://goo.gl/AbAqZY>
- [33] OpenCV, *undistortPoints*. [En línea]. Disponible en: <https://goo.gl/kKaFJV>
- [34] OpenCV, *HoughCircles*. [En línea]. Disponible en: <https://goo.gl/hUDzKu>
- [35] Transformada de Hough. [En línea]. Disponible en:
https://es.wikipedia.org/wiki/Transformada_de_Hough
- [36] OpenCV, *GaussianBlur*. [En línea]. Disponible en: <https://goo.gl/BQ1s7w>
- [37] John Canny, “A Computational Approach to Edge Detection”. doi:
10.1109/TPAMI.1986.4767851
- [38] Ron Brinkmann, “The Art and Science of Digital Compositing”. [En línea]. Disponible en: <https://goo.gl/MztwpG>
- [39] OpenCV, *cornerMinEigenVal*. [En línea]. Disponible en: <https://goo.gl/yjwXeZ>
- [40] Operador de Sobel. [En línea]. Disponible en:
https://es.wikipedia.org/wiki/Operador_Sobel

- [41] OpenCV, *normalize*. [En línea]. Disponible en: <https://goo.gl/4Y2FjM>
- [42] OpenCV, *circle*. [En línea]. Disponible en: <https://goo.gl/SMMZgr>
- [43] Blob. [En línea]. Disponible en: https://en.wikipedia.org/wiki/Blob_detection
- [44] OpenCV, *SimpleBlobDetector::Params*. [En línea]. Disponible en: <https://goo.gl/KWYXoG>
- [45] OpenCV, *triangulatePoints*. [En línea]. Disponible en: <https://goo.gl/Hxm4ty>
- [46] OpenCV, *convertPointsFromHomogeneous*. [En línea]. Disponible en: <https://goo.gl/Tjwh5y>
- [47] OpenCV, *transpose*. [En línea]. Disponible en: <https://goo.gl/5YmGWc>
- [48] Basler Lens Selector. [En línea]. Disponible en: <https://www.baslerweb.com/en/products/tools/lens-selector/>
- [49] Meshlab. [En línea]. Disponible en: <http://www.meshlab.net/>
- [50] Markus Rumpler, Arnold Irschara y Horst Bischof, “Multi-View Stereo: Redundancy Benefits for 3D Reconstruction”. [En línea]. Disponible en: <https://goo.gl/kt3kXt>